
12 Plant Optimization

MICHAEL RUEL

PARTIAL LIST OF SOFTWARE SUPPLIERS

ABB (<http://www.abb.com/>)
Emerson (<http://www2.emersonprocess.com/>)
ExperTune (www.expertune.com)
Invensys (www.invensysflowcontrol.com)
Matrikon (www.matrikon.com)
Honeywell (<http://hpsweb.honeywell.com/>)
PAS (www.pilzusa.com)
Techmation (www.techmation.com)

PROCESS OPTIMIZATION

Rationale

Global markets have forced many factories to rationalize their operations. World overproduction has enabled consumers to become very demanding on quality, defining the desired characteristics of the finished product with precision. In olden days, the measure of success was only the quantity, which is over and done with. Large-scale changes are occurring.

Optimization of an industrial process usually means an increase in profitability while maintaining safety and product quality. The criteria for optimization vary with the process, and the link between product quality and control loop performance is not always clear.

Plant-wide process optimization involves the systems and strategies required to control an entire plant according to control objectives. A plant consists of many unit operations, and control objectives can be in conflict.

Assessing the performance of control systems is not simple, but is now possible; analyzing the process and the control system can be done using normal operating data.

One of the most challenging tasks that a process control engineer faces is how to optimize the processes. For this, the following questions must be answered:

- How do we tune the control loops and the systems?
- Given control objectives, restrictions, and operating procedures, how do we proceed to optimize each unit?
- How do we optimize the entire plant?
- What issues do we need to consider?
- Do we need to model the system?
- Are the strategies appropriate?

- Do we need advanced control strategies?
- Do we need to tune all loops? If so, in what order?
- When optimizing a unit, do we leave the loops in manual mode? In automatic mode?
- Which steps should be done first?

This chapter will guide the reader through the optimization process. Process performance depends on effective use of process control. Included in process control are transmitters and control systems (logic, algorithm, interlocks, etc.) and also pumps, valves, pipes, and other process equipment.

The reader will discover that optimizing a process is similar to a conductor directing an orchestra: coordinate, accelerate here, slow down there, synchronize other parts, and so on. The “conductor” must not only know about control, but also about the process itself. To optimize a control system, one must understand the process perfectly.

There are options for controlling complex processes. An option is the application of simple control systems. Another is as promoted by many academic researchers is the multivariable control systems. The reality is that most of these complex systems may not be fully functional as some sections may be turned off. A common mistake is to use advanced multivariable control with bad equipment or with loops not tuned properly to obtain the expected performances. We should move to multivariable systems when simple techniques fail even if the system is properly tuned.

When optimizing, it is important to consider design, equipment performance, control strategies, operating procedures, performance monitoring, logic, and special control strategies for start-up, shut-down, grade changes, and abnormal conditions.

There are many reasons to use process control:

- Quality
- Safety
- Efficiency
- Improved uptime
- Reliability
- Labor cost
- Operability
- Environmental regulations

Control objectives depend on the goal for the unit.

Impacts of Plant Organization: On Tuning

A plant consists of units connected together, mainly in series. When an entire plant is to be optimized, it is divided into units and each is optimized before looking at the big picture. Each unit is then divided into loops and systems. As an example, surge tanks are explained in this section.

A Plant Is a Series of Units If, say, a surge tank is used between two units, they become almost independent of one another. Each unit should have its own control objective and performance criteria.

Units in Parallel If units are in parallel, it could be useful to synchronize them to ensure that the reaction time is similar in each. Again, the use of surge tanks will reduce interaction.

Recirculation If a part of the output from a unit is recirculated, then the dynamic is different. The plant-wide control problem becomes more complex and its optimization less obvious. If recirculation varies too much, then one should consider changing the tuning parameters.

Recirculation is used to

- Modify the process time constant
- Improve yields
- Improve thermal dissipation
- Improve control response
- Operate the equipment at maximum efficiency

Some of the impacts of increased recirculation are as follows:

- The equivalent process gain is reduced.
- The equivalent time constant is reduced.
- The equivalent dead time is almost unaffected.

Figure 12.1 shows an example of a simple process with recirculation.

Process Optimization Plan

Sustaining Results A good optimization plan requires a methodical approach. Traditional methods can be used; however, modern tools involving computers and software should be favored. More important than the tools, the approach used

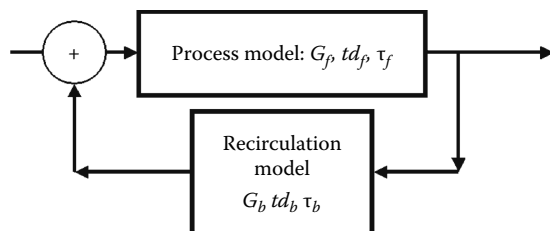


FIG. 12.1
Simple process with recirculation.

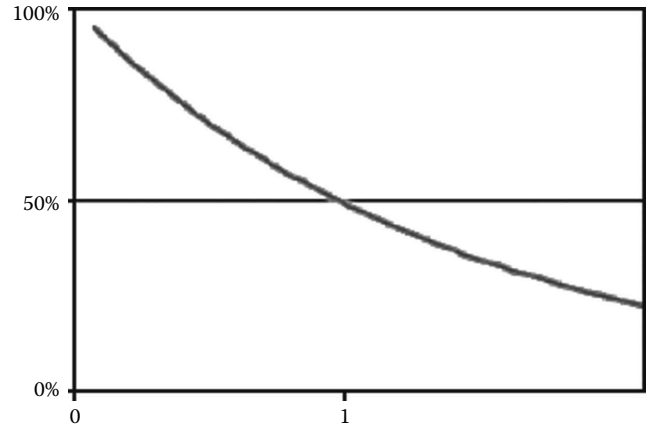


FIG. 12.2
Performance decrease by 50% after 6 months if no effort is made.

must ensure that constant efforts are exerted to achieve and maintain performance improvement. The performance of process control systems will decline if the system is not optimized on a regular basis. In most plants, this decline will result in a performance improvement decrease of 50% every 6 months as charted in Figure 12.2.

Often during start-up, when a loop is placed in automatic mode, if it appears to work (not perform), that loop will be forgotten because there are more important issues. There is no time later to refine the tuning. When a manager decides to optimize this unit, then finally the design will be verified, the equipment will be checked, and the controller will be tuned. However, this optimization effort should be an ongoing and continuous process.

The level of support required to maintain process optimization is a function of process complexity. The plant management first needs to decide whether they have all the expertise and the time to do it internally. In most cases, even if the expertise is there, time will be an important issue. In the event the senior staffs decide to do it using internal resources, they need to make sure that their people have the necessary skills. They also need to use the most modern tools to achieve results efficiently. For example, trial-and-error tuning methods are not the best way to achieve performance. The first step that the plant management will need to take is to train their employees.

The staff must be experienced and motivated. External resources combined with key people are always essential when consultants work in an open manner and share their know-how with plant employees.

Budget and Workforce For a typical plant, the workforce should consist of trained people using proper tools:

This does not include workforce to calibrate, repair, program, etc.

In a typical plant, 20% of the loops have a direct impact on product quality. These loops should be optimized twice a year or their performance should be closely monitored.

- 8 loops/day/person
- 2 times/year
- 240 days/year
- ~1000 critical loops/person/year

Other loops should be verified every 2 years or each time a process change occurs: starting from flowsheet modifications, equipment replacements, resizing, etc.

- 8 loops/day/person
- 0.5 times/year
- 240 days/year
- Here ~ (4000 loops/person/year)

Using a control performance monitoring software package and optimization tool, one person can coordinate loop optimization for a large plant, and it will not necessitate a full-time job.

However, these numbers vary according to the type of industry, process control system, control system strategy, and so on.

The price for software, hardware, networking, and configuration can vary greatly; \$100 per loop is considered a good average. If amortized over a 5-year period, this corresponds to \$20 per loop per year.

The cost of administering the program and providing training could represent \$50,000/year for a large plant and \$20,000/year for a small plant. On a per-loop basis, this could represent \$10–\$40 per year. The total cost is less than \$100/year per loop.

Process Optimization Benefits Process optimization consists of the following:

- Defining performance objectives: This has to be done with the participation of all the departments concerned:
Operations
Management
Maintenance
Engineering
Eventually, marketing and sales
- Analysing the process, control strategies, equipment, operating procedures, and maintenance procedures, and determining their performance
- Fixing the problems encountered, such as transmitter location, valve installation, process gain (linearity), equipment failure, etc.
- Defining a tuning strategy
- Tuning loops and implementing advanced control strategies
- Measuring the performance
- Producing a report
- Planning the follow-up

Process optimization ensures several benefits:

1. Fewer process upsets: With controls properly tuned, there are fewer process upsets. The control loops can

do what they were designed to do: maintain consistent process results.

2. Smoother process start-ups: Process start-ups will be smoother and more consistent. Operations can quickly move into full production. With the process under control, there are fewer alarms.
 - Faster grade changes—product or grade changes can be accomplished smoothly and quickly, with a minimum of scrap.
 - Better operation

Each of these benefits reduces the amount of operator attention required for routine operation. This frees the operator to work on other tasks such as improving operation.

Approach to Variability: “Variability Is Not Removed; It Is Managed”

Variability has been a buzzword for years. It is a relative measure of variance and expressed as a percentage of the mean and so enables comparison between different processes. Based on a normal distribution, the variability of data represents the area which includes 95% of all points, the average ± 2 standard deviations.

In a process, the goal of the process control system is not to remove the variability; it is to move it where it should be directed. Therefore, we do not reduce or remove variability, we manage variability.

Consider a *drying process* in which a continuous strip of material is moving through a dryer section where the energy used is steam. The goal is to deliver the end product at constant moisture content using minimum energy and remaining inside the limits specified even if the moisture content of the material coming to the dryer varies.

As the rate of production varies, the total amount of energy needed has to change. As the temperature of the material varies, the total amount of energy needed has to change. In addition, all along the process, other disturbances occur: ambient conditions, quality of steam (the energy content depends on pressure and temperature), etc. To ensure that these disturbances will not affect the moisture content of the end product too much, one must tune the controllers accordingly to attain this goal.

If variability is reduced on moisture content, the set point (SP) can be increased to control closer to the acceptable limit. Not only will the product uniformity be better, but also the amount of energy and the amount of dry product sold will be reduced; fortunately, because the price per ton at standard moisture content remains the same, the profit margin will be increased. [Figure 12.3](#) illustrates the results before and after optimization.

Before optimizing the dryer, operators had to reduce the SP to ensure that the end product would stay below the moisture-content high limit; hence, the average moisture content was lower than desired, and too much energy was used to dry the product. After optimization, the SP can be increased closer to the constraint.

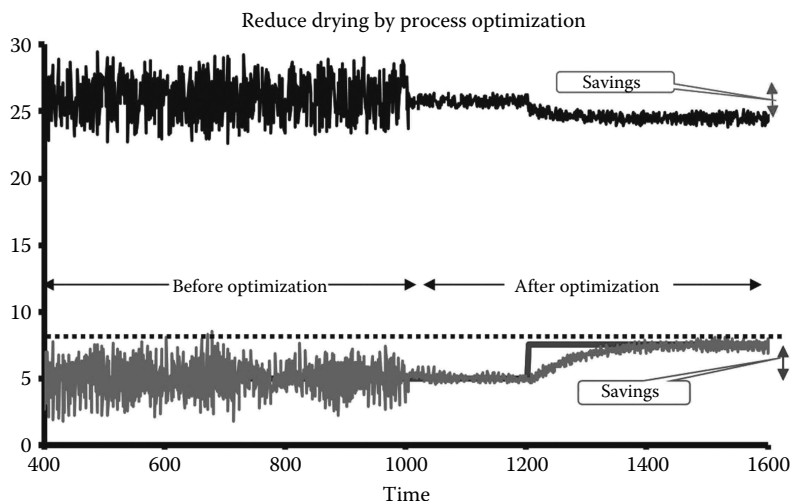


FIG. 12.3
Dryer optimization.

The benefits of optimization are as follows:

- Less energy is used to dry the product, and therefore costs are reduced.
- Less variability in moisture content, better quality.
- Less material sold, more water, and costs are reduced.
- Valve life expectancy is increased because valve movements are reduced.

PROCESS OPTIMIZATION METHODS

Loop Tuning

Process Optimization Is More Than Loop Tuning Loop tuning is different from process optimization. To optimize a process, it is essential to work on the overall picture. To do so, one must organize its work and tune each loop according to its function in a set of interacting loops.

Process and control system audits assess the performance capabilities of operating units to determine where improvements can be made and to compare their performance with industry standards. An audit is done by observing many variables at the same time to determine whether the performance criterion for each loop is being met. From an audit, it is possible to detect cycling problems, instability problems, and operating problems.

Auditing the process is time-consuming and is usually done only when major problems are present. Using software, it is now possible to audit the process continuously. Software looks at the data from the process and computes performance indices at regular intervals. If a threshold is reached, an alarm is triggered. The program identifies the areas of the plant where the greatest economic gains are possible. The program is designed to help plant personnel make the greatest impact on the plant. It pinpoints the areas that will yield the greatest economic return.

Optimization Plan A good optimization plan should include and define

- Budget
- Plan
- Training
- Software and other tools
- Data acquisition strategy
- Schedule
- Work plan: auditing, tuning, fixing problems, etc.
- Reporting
- Follow-up
- Justification
- Return on investment

Process optimization concerns include

- Quality improvement
- Security
- Cost of material
- Grade change
- Reduction in maintenance costs

Process Optimization Process optimization consists of adjusting each loop and system to match its control objective while at the same time reaching the main goal for the process.

Process Models Software packages which are available to tune loops and to optimize a process use time-series and frequency-analysis techniques. The effectiveness of any control system depends largely on how well the process under control has been mathematically modeled. Some control designs are less sensitive to model error than others. However, as control performance requirements increase, generally so does the requirement for model accuracy.

Control, instrument, and process engineers deal with many types of process models. Each type of model has

TABLE 12.1*Types of Process Models Used in Control Work*

<i>Computer Model Type</i>	<i>Typical Application</i>
Large-scale nonlinear dynamic simulation	Used for off-line investigation of processes, process dynamics, and control applications. (Large effort required to generate models.)
Frequency response models	Used to investigate open-loop behavior and to specify closed-loop feedback control. Used to calculate tuning parameters.
Single-input single-output (SISO)	MIMO models are used to study interactions. (Model typically generated from state-space models or plant testing.)
Frequency response models	
Multiple-input multiple-output (MIMO)	
State-space models, MIMO	
Continuous and discrete linear models	Used for control system design with methods such as linear quadratic optimal and pole placement control. Used for estimation and predictive models. (Models typically generated from linearization of simulation models or plant tests.)
Time-series models, SISO	Used for adaptive control and state parameter estimation.
Difference equations	
Simple low-order models	Used for quick time-domain model fits to plant tests for control-loop tuning.
SISO continuous time	Used for quick time-domain model fits to plant tests for control-loop tuning.
High-order models from frequency-response and time-series analysis	
reduced to simple models	

a different use depending on the specific application. Table 12.1 provides a partial list of these computer models and some of their typical applications.

Several of these nonlinear dynamic simulation packages are available commercially. Many of these packages have several common features. First, the simulation models require the input of many differential equations representing the process mass and energy balance equations.

Once the process has been adequately modeled, control is added to the dynamic simulation. The time behavior of the closed-loop system is studied to evaluate the adequacy of the control achieved. The control scheme can be changed to try new, intuitively devised schemes or control designs recommended by other computer-based design methods such as linear quadratic optimal or frequency-response methods.

Performance Indices

Many performance indices can be used to determine whether a goal has been reached:

- Variability (the relative value of twice the standard deviation, expressed as a percentage of the mean)
- Response time, settling time
- Overshoot
- Phase and gain margins
- IAE (integral of absolute error)
- Valve travel and valve reversal, variance of controller output (CO)
- Others: ISE, ITAE, ITSE, Harris index, comparison to minimum variance control, % time not in normal mode, % time at saturation, area in autocorrelation, flat power-spectral density, numbers of alarms per day, % of power at a specific frequency (or a cluster of frequencies) on a power-spectral density graphic, etc.

Table 12.2 gives some criteria for what kind of equipment performance can be considered good or acceptable.

Various studies, as well as our observations, have confirmed that the typical performance distribution of control loops is as in Table 12.3 (this has remained the same for at least the past 10 years).

Even if 25% of loops perform better in automatic than in manual, they do not necessarily perform to their maximum

TABLE 12.2*Criteria for Acceptable Equipment Performance*

<i>Typical Control-Loop Problems and Performance Issues</i>	<i>Loops (%)</i>
Control valves of poor quality or in poor condition	30
Poor controller tuning (unacceptable values)	30
Poor controller tuning (not selected according to the performance goal)	85
Poor loop design	15
Controller in manual mode	30
Control loop not performing according to control objectives	85
Loops that perform better in automatic than in manual	25

TABLE 12.3*North American Reality: Typical Performance Distribution of Control Loops*

Process gain	>0.5 and <2
Hysteresis	<2%
Stiction	<0.5%
Noise	<2%
Positioner overshoot	<20%
Linearity ($G_{\max}/G_{\min}$)	<2

TABLE 12.4
Expected Values for Performance Indices

Performance Measurement	Units	Usual	Ideal	Realistic Target
Variability	%	2–3	0	<1
IAE	% · d	4000	0	<1000
RRT and settling time	s	$20 \cdot t_d$	$\ll$	$<10 \cdot t_d$
Valve travel	%/d	5000	$\ll$	<1000
Valve reversal	N/d	7500	$\ll$	<2000
Power-spectral density, power in a single frequency component (or cluster of harmonics)	Percentage of total power	4	$\ll$	<2
Robustness	Absolute	2	>	>2
Percentage time in normal mode	%	85	~100	99

capacity. In three out of four cases, not only does the controller fail to improve the performance of the finished product, but it also actually worsens it.

Expected Performance Order-of-magnitude values for common performance indices and average values for a loop are given in Table 12.4.

Tuning Parameters To analyze the process and the control systems, various techniques can be used for simple observations and determinations of control parameter values. For example, knowing the type of process, the current values can be compared to typical values; the filter time constant, integral time, and derivative time should make sense (Table 12.5).

The most common problems found are as follows:

- *Tuning too aggressive.* Process variable (PV) and CO trends have a sinusoidal waveform in the time domain and a single peak in power-spectral density is observed.
- *Stiction problem.* The CO trend has a sawtooth waveform in the time domain, and peaks are observed at multiples of the cycling period (fundamental + harmonics) in the power-spectral density plot. The data are not distributed normally (bell curve) on a distribution plot.

TABLE 12.5
Usual Tuning Parameter Values for Common Loops

	P	I	D	F	Settling Time
	K_p	T_i	T_d	τ_F	
Flow, pressure (liquid)	<1	Seconds	0	Seconds	Seconds
Level	>1	Minutes	0	Seconds	Minutes
Temperature		Minutes	Seconds	Seconds	Minutes
Analysis		Minutes	Seconds	Seconds	Minutes

- *Backlash or hysteresis problem.* The PV and CO trends exhibit oscillations, but the period increases as the PV reaches the SP.
- *Interacting loops.* Cycling at the same period in many trends and similar power-spectral density on many signals; cross-correlation analysis will yield strong correlations.

Process Optimization Tools

- Power-spectral density: should be flat.
- Cumulative power-spectral density: should be continuous.
- Statistical analysis: data distribution should be a bell curve, variability should be small, and valve movements should be minimized.
- Process modeling: to validate the process and to find tuning parameters.
- Robustness analysis: to validate tuning parameters.
- Process analysis: hysteresis and backlash, stiction, noise, deviations from process model, hidden cycling, etc.
- Autocorrelation: emphasizes any oscillations; the area under the curve is a good indicator of control effectiveness.
- Cross-correlation and multivariate techniques: to measure interaction between signals and loops; also to determine how helpful multivariable control could be.
- Performance index monitoring: variability, IAE, Harris index, etc.

When problems have been identified, they should be fixed:

- Process: mechanical, electrical, process, operation
- Nonlinear and variable process: a more elaborate control strategy
- Interacting loops: a good tuning strategy is generally enough; if not, multivariable control could help

Control Loop Objectives

- Main goals
- Remove disturbances
- Follow SP
- Reduce variability

The goal of a control system is to maintain the PV equal to the SP in the presence of load changes, disturbances or SP changes. Tuning a controller means adjusting the parameters (i.e., proportional, derivative, and integral, PID) to achieve “Good Control.” “Good Control” is a performance criterion that depends on the application.

We must induce a change in the SP or the load to see the dynamic loop behavior. A loop that looks good in steady state may not behave as desired when changes occur. To determine whether a loop is properly tuned, one must induce a change. The easiest way is by creating an SP change or a load change.

An SP change is more drastic than a load change because with an SP change, the error changes instantly. A load change must pass through the process and is more gradual. Moreover, unless the load change is measured, it is not possible to model the process on the basis of a load change.

Simply looking at steady-state data will determine whether a loop is stable; to know whether a performance objective has been reached, we must observe the response following an SP change or a disturbance.

Software tools to monitor performance will analyze the process under normal operation using:

- Time-series analysis: for example, autocorrelation analysis will yield the impulse closed-loop response and will estimate the settling time.
- Frequency-response modeling: for example, to verify loop behavior.
- Multivariate statistics: for example, to analyze stiction.
- What is good control? One must ask the following questions:
 - Is it fast?
 - Is it without error?
 - Is it without overshoot?
 - Does it minimize variability?
 - Is it without oscillations?
 - Does it minimize errors? Is it tight?

Tuning objectives will normally vary from loop to loop. Very fast control with a slight continuous oscillation may be fine for one loop, but dangerous for another.

For example, the controller of a robot arm moving a bottle filled with nitroglycerin will be tuned without overshoot and without oscillation. The response will be slow and sluggish (Figure 12.4).

On the other hand, a flow controller in a loop filling milk bottles will be tuned tightly and aggressively to ensure the maximum speed of the production line with minimum error

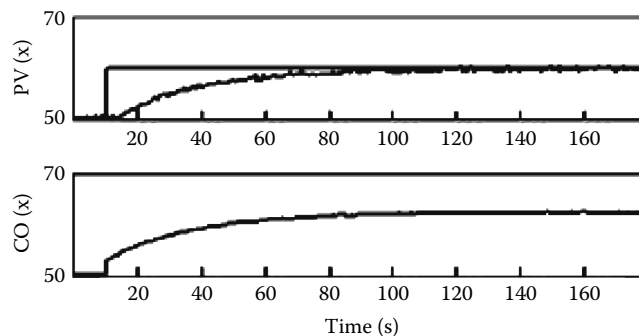


FIG. 12.4

Loop tuned for critical damping.

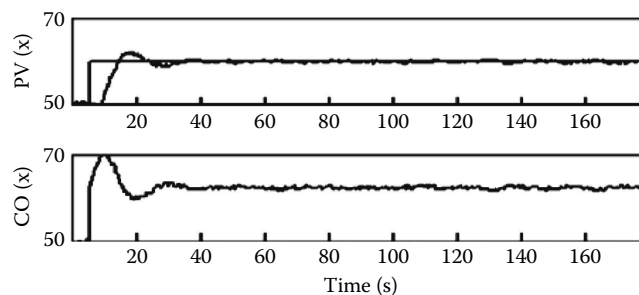


FIG. 12.5

Loop tuned aggressively.

(Figure 12.5). In such a system, the process tolerates oscillations but must minimize the error.

Tuning a controller not only makes the process operational, but also makes it meet the performance criteria.

Common Performance Criteria Figures 12.6 through 12.9 illustrate different performance criteria.

The least aggressive tuning criterion is called “critical damping” (Figure 12.6). It gives the quickest possible response without overshoot. Trying to go even slightly faster than critical damping will create overshoot. For this reason, processes that can tolerate no overshoot will normally use even less aggressive tuning, just to be safe. A method developed for this purpose is called “lambda tuning” and is extensively used in the paper industry. The closed-loop time constant is usually chosen as three times longer than the process time constant. When the criterion is variability reduction, the goal is to remove a disturbance without overshoot or cycling in an effort to minimize variability. For example, when producing paper, the variability is embedded into the sheet of paper, so it must be minimized everywhere. In contrast, when producing fuel gas, because the product goes into a large, mixed storage tank, the variability is averaged by mixing.

The second response has an overshoot H (Figure 12.7). It is normally expressed as a percentage of how large it is compared to the steady-state value ($\text{Percent Overshoot} = H/C \times 100\%$). Common tuning goals are 5% or 10% overshoot. This tuning gives a fast response, but some oscillations are to be expected.

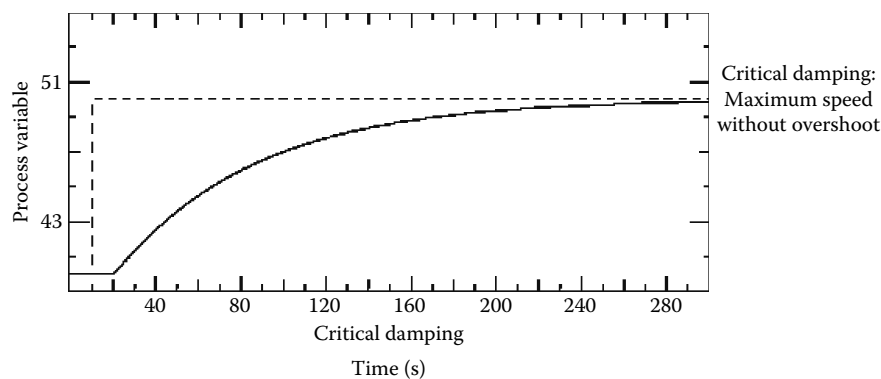


FIG. 12.6
Critical damping.

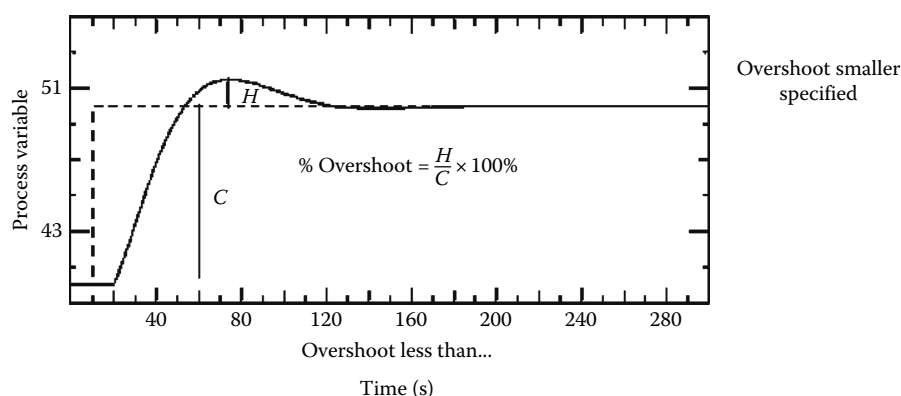


FIG. 12.7
Overshoot less than a specified value.

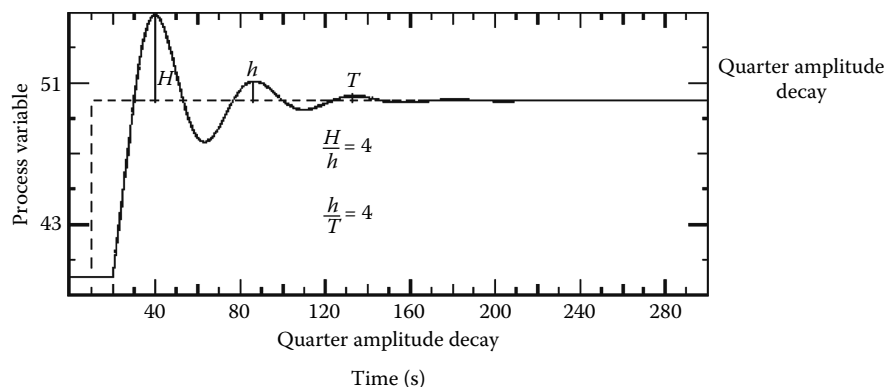


FIG. 12.8
Quarter-amplitude decay.

The third and fourth responses are “Quarter-Amplitude Decay” (Figure 12.8) and “Minimum Integral of Absolute Error” or “IAE” (Figure 12.9). These two have a very similar shape and give very quick response with obvious oscillations.

Quarter-amplitude decay specifies a damped oscillation in which the magnitude of each successive positive peak is one-fourth that of the preceding positive peak. This criterion is popular because it is easy to understand and easy to apply in the field. It differs from IAE in that IAE will be slightly quicker and will have slightly less error (PV compared to SP).

A faster response than IAE is not practical because it will entail a greater total error and more overshoot and will take longer to stabilize. Loops tuned to have quarter-amplitude decay or IAE response should be retuned more frequently than others, because any error in tuning or any changes in the process will give a highly oscillatory response that could become unstable. This criterion leads to non-robust tuning parameters. Tuning for tight control is aggressive, with the goal of minimizing IAE. A pressure loop where the SP is close to the relief valve SP pressure adjustment is a good example.

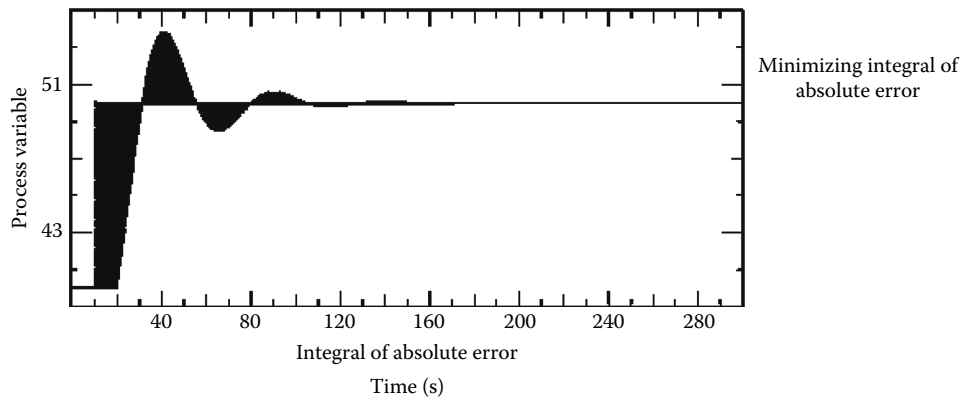


FIG. 12.9
Integral of absolute error.

A good compromise is tuning with 10% overshoot. This response is quick with very little or no oscillation. This tuning also gives peace of mind. If the process dynamics change slightly, the response may become more aggressive, but it will not become unstable, as may occur with IAE or quarter-amplitude decay; the tuning parameters are also more robust.

There is a relationship between these responses that is very useful when tuning loops on the spot. Reducing the controller gain K_p by a factor of 2 will take the response from aggressive to moderate. Reducing it by half again takes the response from moderate to slow. It is important to note that this is only the case with ideal and series-type controllers. If a controller with a parallel structure controls the loop, the integral and derivative gains must also be cut.

Tuning: A Compromise Tuning a controller consists of making compromises between speed and stability (Figure 12.10). The more stable the loop is, the more slowly it reacts, and the larger are the errors after a disturbance.

In industry, control loops are nonlinear (the process model is not constant), and it is essential to tune them for the worst case. When possible, if the process model varies widely, the control algorithm should also vary.

The more tightly and aggressively the loop is tuned, the more it will react quickly, but the price to pay is instability, cycling, and overshoot.

Before tuning a loop, you must ask the following questions:

- Is the process slow or fast?
- Is the process self-regulating?

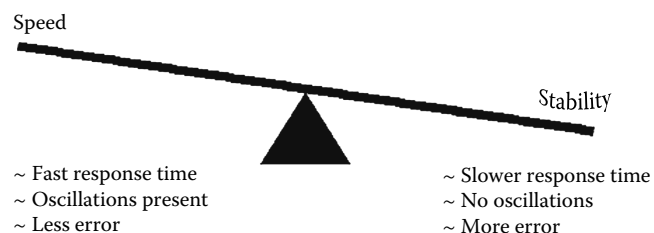


FIG. 12.10
Tuning, a compromise between stability and speed.

- Is the process symmetrical?
- Is the process linear; is the right valve characteristic used?
- Is the process gain near one; is the valve properly sized?
- Is the transmitter properly installed and working well?
- Does the valve have hysteresis?
- Does the valve have stiction?
- Is dead time dominant?
- Is noise excessive?
- Will the tests represent the worst cases?

After tuning, the tuning parameters will be valid until a change is made to the process or to a component in the control loop.

The role of a control loop is, most of the time, to maintain the PV at SP. To achieve this, the CO modulates a final control element to reject disturbances. For some loops, for example, the inner loops of a cascade control system, it is also essential to obtain a good SP response. Finally, in a batch control system, the PV must follow the profile SP, generally without overshoot.

Hence, when optimizing a process control system, the process control engineer must consider the process control objectives for the loop, the unit, and the entire plant.

How well each controller accomplishes its task depends on the selection of the controller parameters according to the process and the control objectives; this is loop tuning.

Single-loop tuning is covered in other parts of this book. Most of the time, a well-designed PID controller is still the best algorithm to handle the job. Various other algorithms are available for difficult processes.

Data for Tuning When analyzing data, the first question is, how good are the data? To answer this question, the following should be considered:

- Sampling rate: fast enough?
- Between points: snapshot or averaging?
- Report by exception (report change only if the change is above the threshold)?

- Data compression?
- Filtering?
- Quantization?
- Integer or floating-point numbers?

Other functions applied:

- Scaling
- Filtering
- Averaging
- Characterizer
- Limiters (amplitude, rate)
- Interlocks
- How good are the configuration and the programming?
 - Scan rate
 - Program sequence
 - PID function and selection
 - Scaling in PID function
 - Algorithm
 - Bumpless transfer
 - Filtering, averaging, moving average
 - Derivative on PV or on error
 - Anti-reset wind-up active
 - Dead band
 - Rate limiters, SP filter
 - Integer
 - Limits on integral, derivative, unit multipliers, and quantization

CONTROLLER ALGORITHMS

Filters

For most loops, a simple PID controller is used. In fact, a filter should be used most of the time to remove noise. Noise corresponds to fast disturbances, which a PID controller cannot handle. The filter will not remove the noise, but will hide it from the controller. Hence, the CO will be noise-free, and this will reduce valve movement.

Characterizers

Sometimes, the simple PID controller needs help to ensure that good performance will be achieved. If the process gain varies, performance will vary as well. To maintain good control under all conditions, the controller must behave the same way at all rates and under all conditions. To ensure that it does, it could be useful to make the following changes:

- Add a characterizer at the CO, to change the valve characteristic, for example
- Add a characterizer at the input (PV and CO); pH loops are an example

- Modify the tuning parameters based on process conditions or on a variable, for example, for grade changes
- Modify the control algorithm, as explained in the next paragraph

A standard PID controller is normally implemented using one of the three most common structures: ideal, series, or parallel. Although these structures differ, the behavior of all three is similar; the input to the controller is always the error (PV-SP or SP-PV), and the CO is composed of a proportional, an integral, and a derivative contribution.

This standard algorithm is a universal solution. It works on all processes, but it does not always provide the desired performance. To increase performance, especially on nonlinear processes, the standard algorithm can be modified. Special algorithms change a standard PID controller and make it nonlinear when some facet of control needs improving.

Nonlinear Algorithms A nonlinear controller is usually used to compensate for process nonlinearities. For a linear process, standard PID control is a better choice than a modified algorithm. However, when the situation warrants a modified algorithm and if it is implemented properly, the result is improved performance. Some controller manufacturers make these algorithms available as a choice in their controller setup. If they are not available, it may be possible to program their behavior into a programmable controller or a DCS.

Standard algorithm. The implementation shown in Figure 12.11 uses an ideal controller structure (also called an ISA structure).

Derivative, D, on PV. An SP change will be handled by the proportional and the integral components only. The derivative is active only when the PV moves. This algorithm is recommended any time that the derivative is used and the algorithm can be implemented (Figure 12.12).

D and Proportional, P, on PV. An SP change will be handled by the integral component only. As a result, there should be no overshoot on an SP change. The proportional and derivative are active only when the PV moves. This algorithm enables more aggressive tuning (Figure 12.13).

SP filtering. An SP change will not provoke as aggressive a response as in the previous algorithm (Figure 12.14). Overshoot on an SP change will be less, and the proportional gain can be increased as a result. This algorithm enables more aggressive tuning and is recommended on loops that have frequent SP changes and disturbances. A filter is preferred over a ramp limiter.

P²ID (error-squared). An error-squared controller will act quite aggressively when the error is large and quite sluggishly when the error is less than 1%. At 10% error, the input to the PID is 100%. At 0.5% error, the input to the PID is 0.25%. This algorithm can give aggressiveness and steady-state stability when properly implemented (Figure 12.15). The most common uses of this algorithm are for level control (averaging and surge).

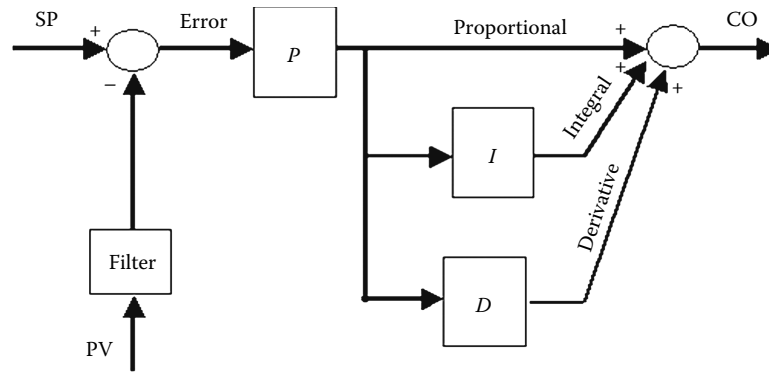


FIG. 12.11
Ideal or ISA controller.

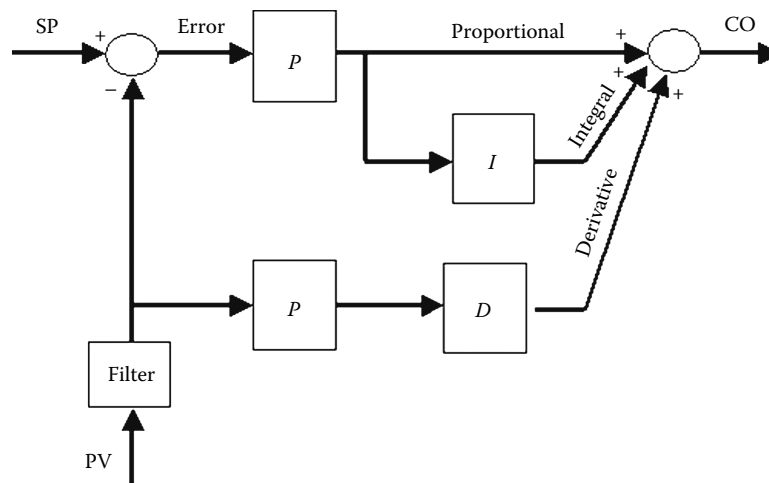


FIG. 12.12
Derivative on PV.

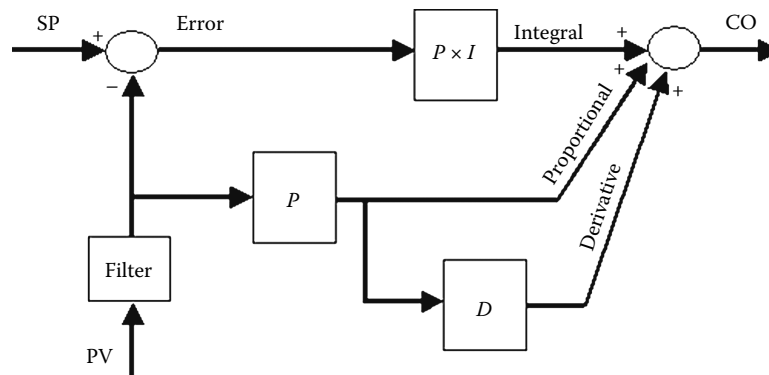


FIG. 12.13
Derivative and proportional on PV.

PPD (integral squared). A less commonly used algorithm is the PI^2D (Figure 12.16). Because integral action is normally linked with dead time, it makes sense that integral-squared action works best to compensate for processes with a variable dead time. Integral action causes overshoot in level loops; with integral-squared, as the level approaches SP, the integral contribution goes to zero.

PID + gap. When the error is within the gap, it is considered to be zero. Outside the gap, the normal tuning parameters apply. Using this algorithm, the CO changes are reduced, increasing the terminal-element life expectancy (Figure 12.17).

PID + gap + adaptive tuning. To fight stiction, when inside the gap, integral action is removed; to fight hysteresis,

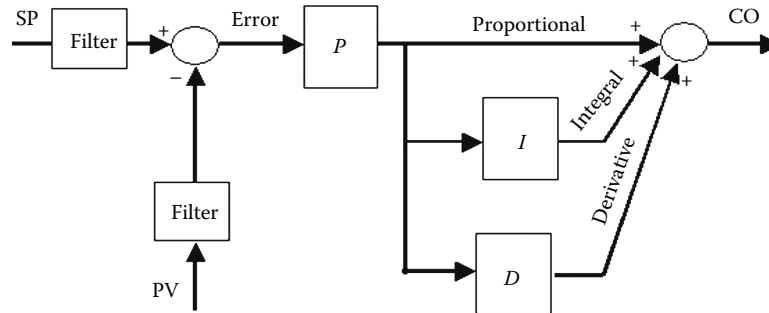


FIG. 12.14
SP filtering.

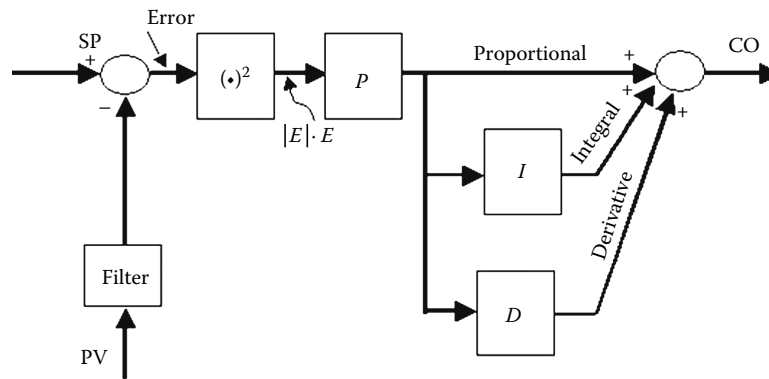


FIG. 12.15
 P^2ID algorithm.

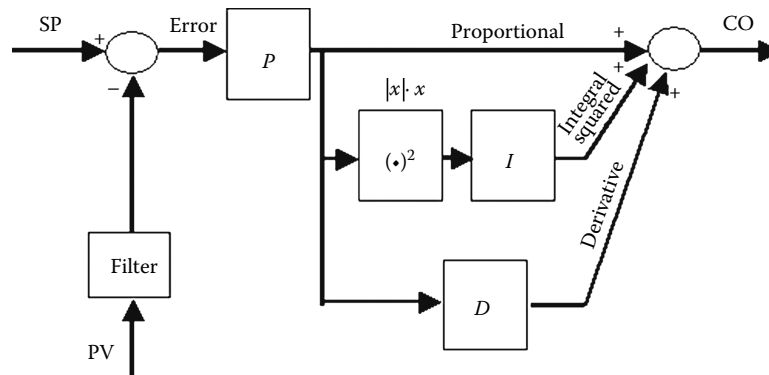


FIG. 12.16
 P^2D algorithm.

when inside the gap, the proportional action is reduced; to fight nonlinearity, when inside the gap, proportional action is reduced.

Many algorithms have been suggested to replace the PID controller, some of which are

Smith predictor. This controller gets its feedback from a model of the process without the dead time, rather than from the real process. The real PV is compared with models to correct errors and handle disturbances. This controller is good for processes with large dead

time that have SP changes; it features no overshoot on SP changes, and it is as fast as or only slightly faster than normal PID control on load changes.

Dahlin controller/model predictive controllers. The controller behaves as one over the process [$process \times (1/process) = 1$], giving a closed-loop response that has the same form as the open-loop response. This is similar to lambda tuning: beneficial for loops that can tolerate no overshoot on an SP change. There is no overshoot on SP changes, and it is very sluggish on load changes.

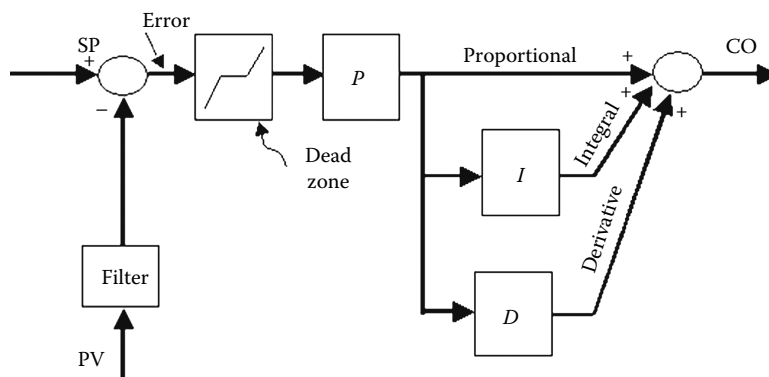


FIG. 12.17
PID + gap algorithm.

Adaptive model controller. This controller is the same as a model predictive controller except that the model is continuously updated. It is very good in mechanical systems, but does not handle process problems very well. Hysteresis, stiction, and nonlinearities cause the system to “remodel” continuously.

Model-free controller. At the time of writing, there have been no model-free controllers that truly work without any prior knowledge (or approximation) of the process model.

MULTI-LOOP AND INTERACTING SYSTEMS

Unfortunately, when there are many loops in a system, there is no magic formula that will tell you whether one loop will affect another. This information is available only through knowledge of the process. If one loop directly feeds another, oscillation in the first loop will cause oscillation in the second and possibly any other downstream loops. If two flow loops are fed from the same pump, oscillation in one loop could cause oscillation in the second. At other times, loops do not interact, but it is imperative that they respond with the same speed.

Interacting Loops

Figure 12.19 shows a solution flow loop. Two liquids are brought together. The top flow is an expensive product and is maintained at 100 USGPM. The lower flow is water; it is adjusted to give the total flow required (between 200 and 400 USGPM).

All three of these loops have the potential to be fast. A response time of less than 30 s is attainable on all three loops, but which loop *needs* to be fastest?

In this case, the pressure loop must be faster than flow loop 1; its job is to maintain pressure for the flow loop, regardless of what the flow is. If the pressure loop and flow loop 1 are tuned at the same speed, they may work for a while, but eventually a disturbance will occur that will cause the two loops to oscillate. For example:

1. A disturbance causes the line pressure to increase.
2. If the pump is too slow, the flow will increase.
3. The valve closes to compensate, causing the pressure to increase.
4. The pump eventually slows down, causing the flow to decrease.
5. The valve opens to compensate, causing the pressure to decrease.
6. The pump speeds up, causing the flow to increase.

Steps 3, 4, 5, and 6 repeat continuously—the two loops will oscillate and potentially resonate.

Similarly, because flow loop 1 feeds flow loop 2, flow loop 1 must be faster than flow loop 2. Otherwise, a disturbance in flow loop 1 could cause both flow controllers to react, and oscillation would result.

When loops interact, we need to select response speeds that differ. Using the exact same speed could cause oscillation. Speeds that differ but are close have the potential to create oscillations. To be safe, we should choose response speeds that differ by a factor of 3–5. If a speed difference of less than three is selected, the loops may 1 day start to oscillate.

For the system pictured in Figure 12.18, this would mean that the response time of the pressure loop determines the

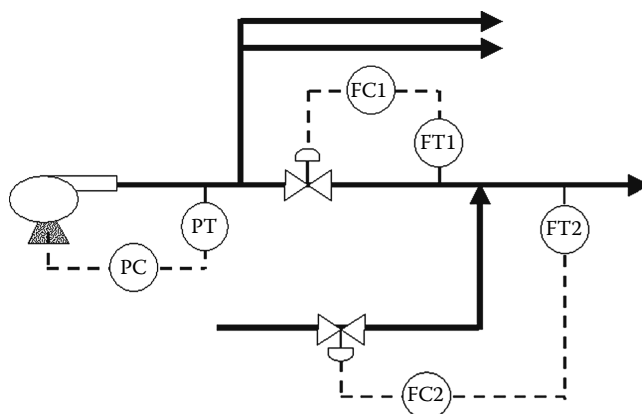


FIG. 12.18
Interacting loops.

response time of flow loop 1. The response time of flow loop 1 in turn determines the response time of flow loop 2. The procedure for tuning this system would be to put the downstream loops in manual and tune the first loop, the pressure loop. It should be tuned aggressively because its response time determines the system response time. Once the pressure loop has been tuned, put the pressure loop in auto, because its behavior is now part of the process. Flow loop 1 is tuned next, leaving flow loop 2 in manual. This loop must be tuned for a response time that is at least 3 times slower than the pressure-loop response time, ideally 5–10 times slower. Once flow loop 1 has been tuned, leave the pressure and flow loop 1 in auto and tune flow loop 2. It, too, must be tuned at least three times slower than flow loop 1. If an optimization software tool handles these interactions, then all loops can be tuned at the same time.

The procedure to tune interacting loops when optimization tools do not handle interactions is the following:

Step 1: **Tune** the fastest loop (PC) for quick response (Figure 12.19); other loops are in manual mode.

Step 2: **Tune** the moderately fast loop FC1 for a moderate response; the previously tuned loop PC remains in auto, while flow loop 2 is still in manual mode (Figure 12.20).

Step 3: **Tune** the slow loop FC2 for a slow response; the previously tuned loops PC and FC1 remains in auto (Figure 12.21).

Synchronized Loops

Sometimes loops do not interact, but they do work together. When loops need to have the same response time, they are called synchronized loops. It is important to note that synchronized loops should be designed so that there is no physical link between them that could cause interaction.

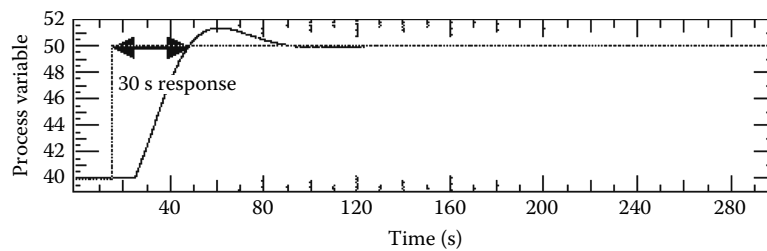


FIG. 12.19

Fastest loop settling time is 30s.

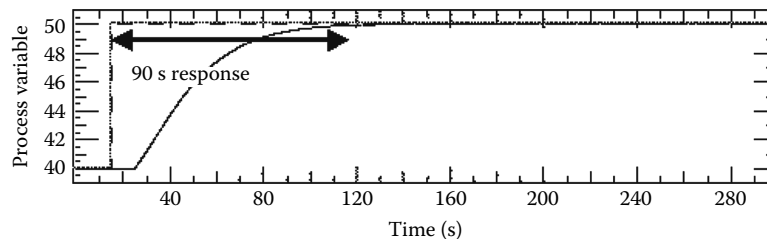


FIG. 12.20

Fastest loop settling time is 90s.

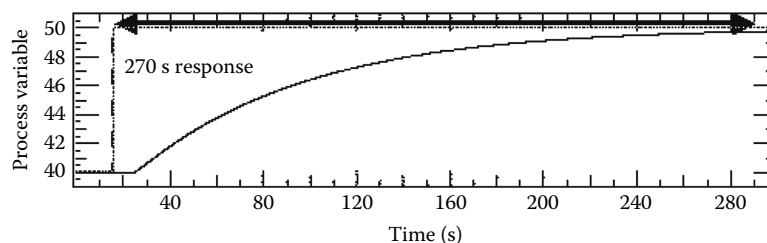


FIG. 12.21

Fastest loop settling time is 270s.

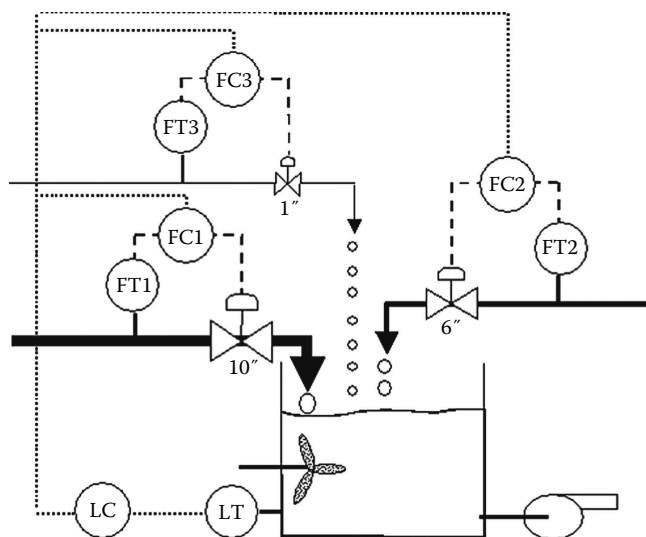


FIG. 12.22
Synchronized loops.

Batch mixing is one example of a system that needs to be synchronized. Figure 12.22 shows three ingredients that are sent into a mix tank. The proper ratio of ingredients must always be maintained, even when the system starts, speeds up, or shuts down.

If the process placed in manual and bumped mode, all three flow loops would probably have different process gains, dead times, and time constants especially when there are large differences in valve sizes. This means that if all three loops were tuned for 10% overshoot, the response times would not be the same. When the level loop calls for increased flows, the recipe will be out of balance until all three flows are stable and at SP.

To ensure that all three loops work in harmony, determine which loop is the slowest and match the response time of the others to it. Normally, the slowest loop is the one with the largest dead time.

Procedure for Tuning Synchronized Loops Model each loop.

Determine which loop is slowest.

Tune the slowest loop aggressively and measure speed of response when done.

Determine tuning parameters for other loops that will give approximately the same response time.

When tuning loops that need to work in harmony, one should select tuning parameters that give similar response times. Using a software optimization tool that can handle multiple loops, this can be done in one shot.

Cascade Loops

A system is in cascade if the output of one controller becomes the SP for another (or others) as in Figure 12.23. The loop receiving the SP is called the inner or slave loop. The loop sending the SP is called the outer or master loop.

The outer process is usually the primary concern and could be controlled by the outer-loop PID only. However, adding inner-loop control can greatly increase performance if the inner loop has disturbances that can be quickly eliminated by the inner loop, but not by the outer loop. For example, the level controller in Figure 12.24 can control the level on its own. However, its response to a disturbance in the incoming flow is poor. By adding a flow controller in cascade, the performance is increased (Figure 12.25) by the ratio of the outer-loop response time over the inner-loop response time (normally a factor of 10 or greater).

Because the cascade loops interact, the inner loop needs to be three to five times faster than the outer loop.

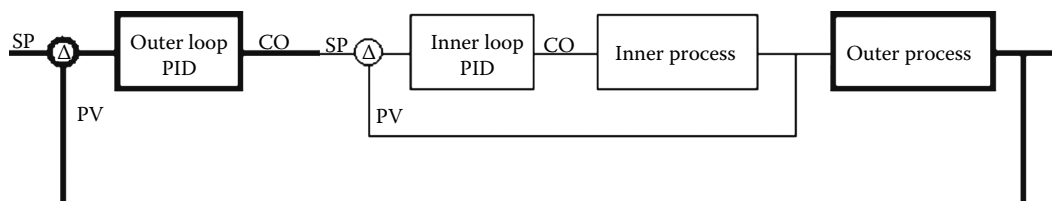


FIG. 12.23
Cascade loop setup.

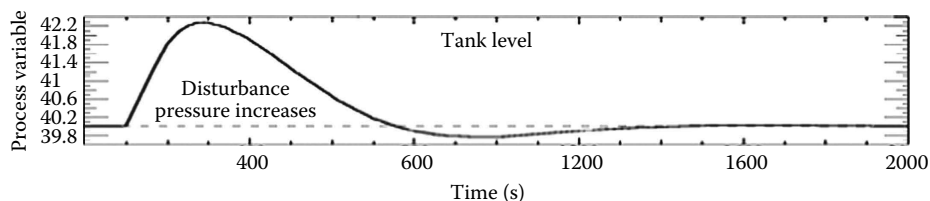
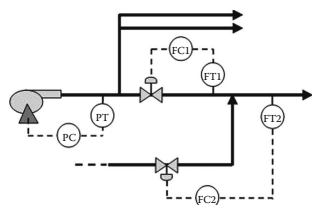
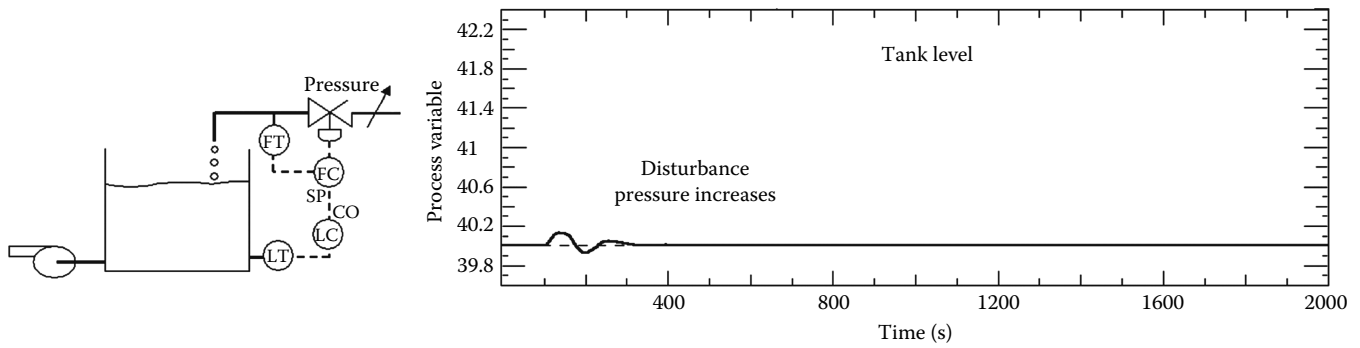


FIG. 12.24
Simple level loop when a pressure disturbance occurs.

**FIG. 12.25**

Cascade level loop when a pressure disturbance occurs.

Tuning cascade loops:

Step 1: Tune the inner loop (see Figure 12.23).

Step 2: Tune the outer loop, leaving the inner loop in cascade mode (auto and remote SP); select tuning to obtain a response three to five times slower than that of the inner loop.

Cascade systems work best when the inner loop can eliminate disturbances quickly. Using a software optimization tool that can handle multiple and interacting loops, this can be done in one test.

Feedforward Loops

Feedforward is a simple but often misunderstood concept. Feedforward is added to “help” a properly tuned PID loop. Without the “help,” the loop will act as any other PID loop, deviating from SP when disturbances occur. When feedforward from a specific disturbance is added, the process should not deviate from SP when that particular disturbance occurs.

Feedforward is added when a disturbance that causes an error can be measured and its effect on the CO is repeatable. By predicting the effect of the disturbance and reacting to it before it affects the process, a huge gain in performance can be achieved. For example, when driving a car on a cold day, if a passenger opens a window, one could act like a PID controller and wait for the temperature to drop before reacting (turning up the heat). If one were to react in this manner, after a few minutes, the temperature would be back in the comfort zone. The better choice would be to turn up the heat as soon as the window is opened, before the temperature drops. This thinking ahead, similar to feedforward control, will ensure that the cabin temperature hardly changes at all.

Feedforward may be added, subtracted, or in some cases multiplied. If the speed needs to be matched, a dynamic compensator (consisting mostly of lead/lag functions) is added.

To add feedforward control, do the following:

First, make sure that the loop is properly tuned.

In automatic, observe the effect of a disturbance on the CO several times (make sure that no other disturbances or changes occur during the test).

Determine the feedforward factor by determining the relationship between $\Delta\text{disturbance}$ and ΔCO .
Test the loop performance with feedforward.

Using a software optimization tool that can handle feedforward control, this can be done in one test, and a feedforward compensator will be calculated, including a dynamic model.

Fuzzy Logic Fuzzy logic is a very interesting topic because its approach to solving problems differs from conventional methods. Whether fuzzy logic is used for process control or to fly a helicopter, one does not need to know anything about the mathematical dynamics of the element being controlled. For example, when a baseball player chases a baseball, he or she does not need to know Newton’s laws of motion to catch it.

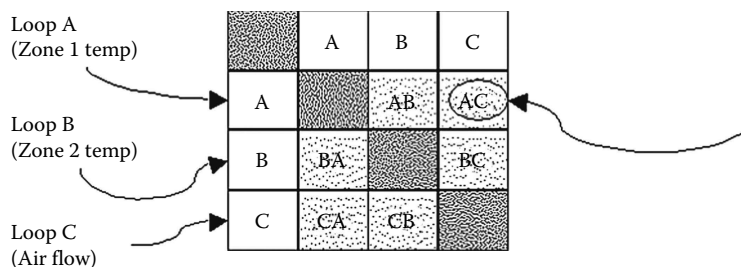
Fuzzy-logic controllers are used to replace an operator when this operator is performing better than a conventional control system; the implementation varies from one process to another. The design consists of transferring knowledge from the operator to the fuzzy-logic controller. By adjusting the rules and the strength of each rule, the fuzzy-logic controller can be made to mimic the operator.

Fuzzy logic can also be used with PID controllers. For example, a PID controller manufacturer has developed an auto-tuner for their PID controller that uses fuzzy logic to determine the optimal PID tuning parameters.

Examples of fuzzy logic in process control:

- As a nonlinear controller (PID replacement): by adjusting rules and fuzzification, the controller becomes nonlinear.
- As a multivariable controller: the fuzzy-logic controller can control multiple loops to cut cost or maximize production. It can maintain tight control of loops that would normally interact with each other.

Neural Networks Neural networks are similar to fuzzy logic in that the mathematical model relating inputs to outputs does not need to be known; the behavioral response is known instead. One major difference between fuzzy logic

**FIG. 12.26**

A 3×3 multivariable system. Each product is an equation that represents the effect of one loop on another.

and neural networks is that we cannot modify the gains or functions in the neural network; we can train the system with data, but we cannot train it with human reasoning. Fuzzy logic, on the other hand, allows modification of each controller parameter.

Multivariable Controllers In a multivariable control system, the control algorithm is designed for the specific system. For example, a model is computed to link all the variables. This is similar to adding feedforward between all loops and potential disturbances in a system. In reality, the controller is a set of equations that enables each component to know in advance the impacts of each move it makes. Hence, the controller will send a new CO to each valve (or a new SP to each loop) each time it makes a move. When properly set up, this enables the entire system to maintain stability when a correction is made.

To build such a controller, a large number of models or equations need to be defined. For example, for a three-input/three-output system, six models are needed (Figure 12.26); with N variables, $N(N - 1)$ models are needed; with 10 variables, you need 90 models. If the number of variables is large, a very complex controller is needed, as well as a large quantity of data, to obtain the models.

Because of the number of relationships that need to be determined, the multivariable controller relatively is expensive to set up. Once it is up and running, it will do a good job until something changes. For example, if the process is redesigned or if equipment starts to wear (hysteresis, stiction, noise, etc.), the controller performance will decrease or become unstable. In many cases, a multivariable controller controls production rates and loop SPs for a whole process area rather than for individual loops. If any one of the individual PID loops is retuned, system performance will change.

The controller links all the control loops and disturbances together; it compensates when required to maintain stability in all linked loops when manipulating a variable. Such a controller is very difficult to implement and test compared to fuzzy-logic and neural-network systems, UNLESS the system is small (3×3 to 5×5). The most important problem is that the system must be retuned when the process changes. Using software optimization tool that can model multiple loops, this can be done in days; traditionally, the modeling of multiple-loop systems required weeks of work.

SOFTWARE TOOLS

Optimization Software

Criteria Many software packages are available to tune or to assist in the tuning of industrial controllers. The criteria listed below are not met by all of them. Most packages are single-loop tuners and are insufficient for process optimization.

Necessary Tools The software should

- Accept ASCII data and/or
- Provide a data acquisition system and/or
- Permit direct connection to the DCS or PLC and/or
- Communicate through DDE or OPC
- Compute the tuning parameters from data obtained in manual or automatic mode
- Compute the tuning parameters from data obtained during normal operation by adding small excitations to the SP
- Include tools for multiple-loop tuning and analysis (cross-correlation, cascade, feedforward, etc.)
- Model the process even in the presence of noise; simulate the process and the closed loop and calculate robustness

Other Tools The software should

- Model interacting loops
- Tune cascade loops in one test; model a cascade scheme and provide tools to tune simultaneously both the inner and the outer loop
- Calculate a feedforward compensator
- Measure interactions and calculate the matrix of various models
- Compute tuning parameters based on user-selectable performance criteria
- Know the structure and characteristics of the industrial controller used (the user specifies the model and manufacturer only), including derivative- and integral-specific methods
- Computes the filter time constants for PV and SP
- Have an excellent support and help file

Many software packages on the market do not meet all these criteria. Some suggest strange values and use incoherent methods.

Closed-Loop Tuning versus Open-Loop Tuning

No More Step Tests Traditional methodology for optimizing and tuning PID loops (excluding “trial-and-error”) relies on “open-loop” tests, in which the loop is placed in manual mode and the CO is moved, usually in a step-wise fashion. The issues with this “step” test are numerous, primarily that the test will disturb the process. This is especially true for slowly moving variables such as temperature, where a seemingly small output move could result in a large process change that could have serious consequences.

There are now tools and methods that enable one to optimize and tune PID loops safely in closed-loop mode. Some of these tools can generate and send small, rapid SP changes to the controller independently of any operator intervention. With the right understanding of these tools, it is now possible to tune every PID loop in a plant optimally with minimal time and risk. If the tests are planned correctly, process control personnel will define the boundaries and conditions and the tests will be done during night shifts or at any other time. Later, the process control engineer will analyze the results and decide on tuning objectives; the software will calculate tuning parameters. Optimizing processes and tuning loops without spending hours in the control room reduces not only the resources needed, but also production losses and disruptions to the attention of operators.

Open-loop tests disturb the process and require the attention of operators. Other open-loop tests can be used in conjunction with software tools, such as so-called pulse and double-pulse tests.

A double-pulse test, when executed properly, can help move a PV back toward SP and shorten the test time. A single-pulse test does not offer this advantage, but shares the benefit of directing the PV back to the original value before it deviates too far from normal conditions.

The issues with “pulse” tests are fewer than with “step” tests, but they still exist. These tests will disturb the process. The loop must also be closely monitored while in manual mode. The complexity of the double-pulse test requires additional skills. Because of these problems and the lack of skills in many plants, many people resort to “trial-and-error” methods in which the tuning values are changed and the response to an SP step change observed. This process is repeated again and again until the response is satisfactory. A little clear reflection should suggest that repeated SP step changes while testing tuning values are potentially more disruptive than a few output pulse changes made in manual mode.

Methodology of Closed-Loop Tuning Any tuning method seeks to establish a “cause-and-effect” relationship between the CO and the PV. To do this in open-loop mode, the output is moved directly. In closed-loop mode, making a change in

the SP causes the output to move indirectly. One key point for either method is that the response of the PV must be due solely to the movement of the CO, whether due to an SP change in closed-loop mode or a CO change in open-loop mode. If the process-variable movement is due to a disturbance, then the test and the data are invalid for determining process models and tuning parameters. The exception to this is if you are able to measure and factor in the disturbance(s) or to make an SP change that causes a process movement larger than that caused by the disturbance. If disturbances occur occasionally, using more data (so that disturbances represent a small portion of the data) will permit the process to be modeled even with occasional disturbances.

Several types of tests are typically used for closed-loop tuning as exemplified in Figures 12.27 and 12.28. Changing the SP up or down in a step-wise fashion is one such test. This test is likely to be as disruptive as an open-loop test unless the SP change is part of normal process operation. The only advantage over an open-loop test is that unless the loop has been tuned to be unstable, it will eventually settle at the new SP, a value that you have selected as a safe eventual target. Changing the SP in a single-pulse or double-pulse test is another type of test. These tests have an advantage over the step test because you return the SP to its normal value. The double-pulse test has a further advantage in that, when done correctly, the PV is forced back toward SP more rapidly than with a single pulse. All loops where enough SP changes are present (with amplitude above noise level) can be tuned using historical data if the frequency of data collection is fast enough.

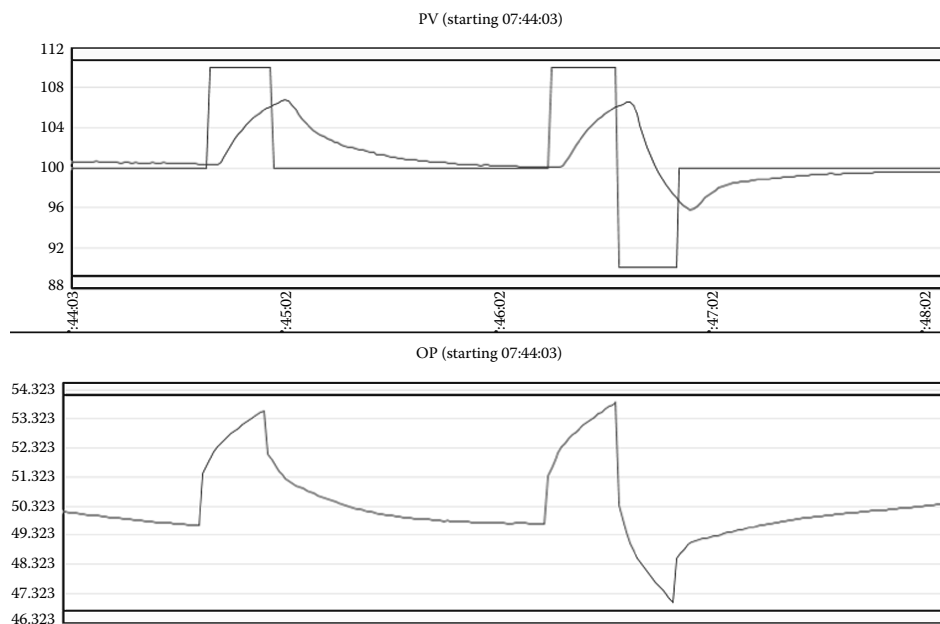
A pseudo-random-binary sequence (PRBS) test (Figure 12.29) is similar to doing multiple double-pulse tests. As with the double-pulse test, the process is maintained at the same average value as the SP, thus eliminating any deviations from normal operation. PRBS is richer and will generate better models than single- or double-pulse tests.

Another possibility is the generalized binary noise (GBN) approach, which is more or less a low-frequency version of a PRBS test. A high-frequency dither is added on top of the low-mid-frequency pattern to improve estimation of dead time (which is a high-frequency component).

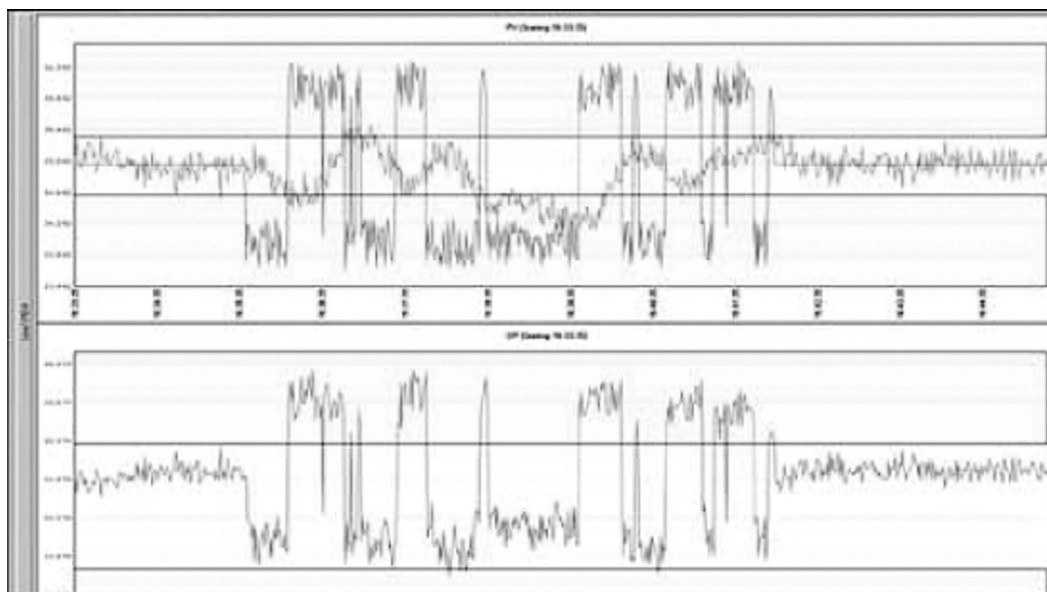
If SP changes occur in a process as a normal part of process operation, these data can be used to model the process and tune the loop. SP changes could originate from operators, APC (advanced process control), or other controllers (cascade strategies, ratio loops, etc.). For example, in a cascade control strategy (Figure 12.30), moving the SP on the master will generate enough changes in both loops (master and slave) to tune them.

When analyzing cascade loops, using the right software will handle secondary and master tuning parameters; if the user changes values or objectives in the secondary loop, the tuning parameters for the master will be automatically recalculated.

Open Loop or Closed Loop Traditionally, loops were tuned in open-loop mode using bump tests. Most tuning tools use this

**FIG. 12.27**

Example of closed-loop pulse and double-pulse tests. The SP and PV are shown in the top plot, and the CO is shown on the bottom. (From Matrikon TaiJi-PID.)

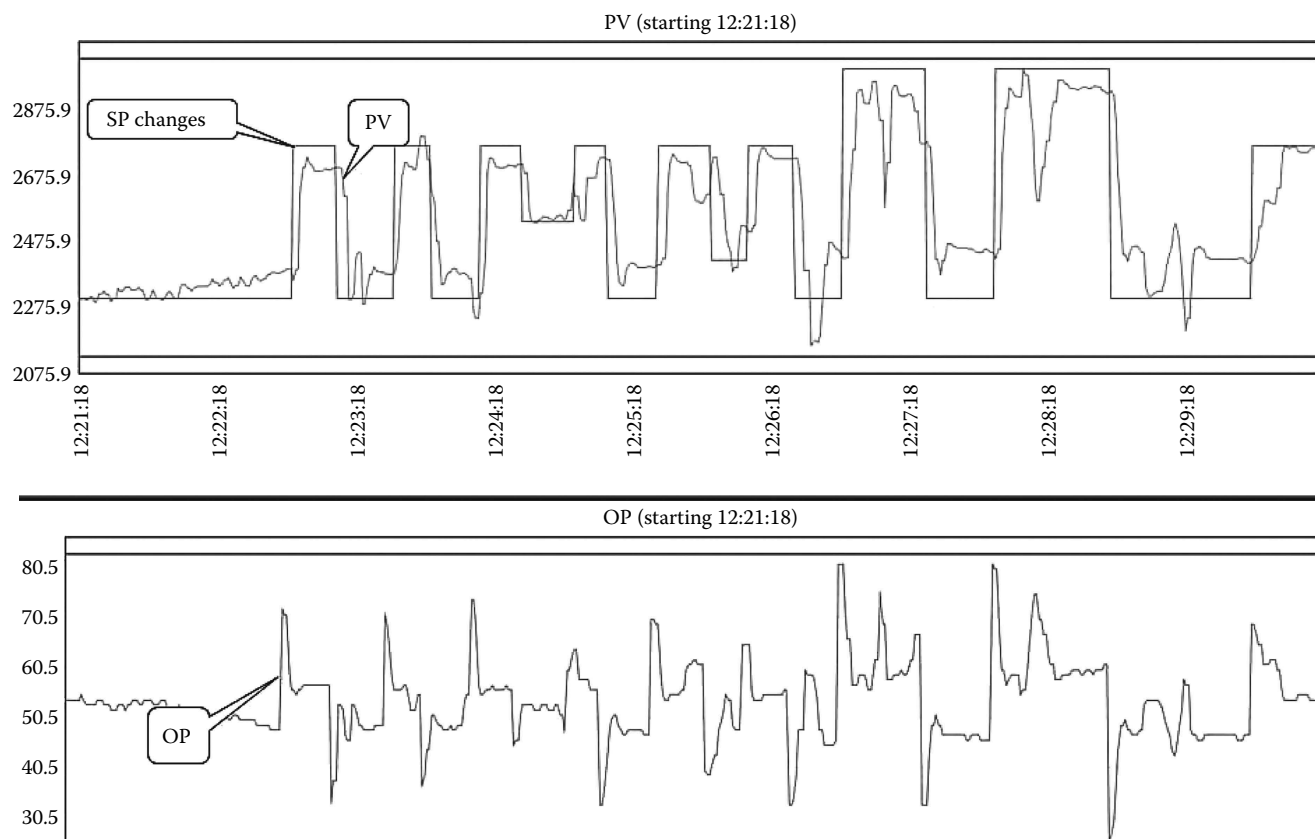
**FIG. 12.28**

Example of an automatically generated closed-loop GBN test with a high-frequency dither. The SP and PV are shown in the top plot, and the CO is shown on the bottom. (From Matrikon TaiJi-PID.)

technique. When analyzing process data in open-loop mode, one must be careful because effects that are programmed into the control system could be missed. For example, controller execution time, filters, ramp limiters, and characterizers will not be included in the test. In addition, simple bump tests require special techniques to detect process defects such as dead band, backlash, hysteresis, stiction, and nonlinearities.

When analyzing a system in closed-loop mode, controller sampling time is included, process defects are part of the model (if the SP moves in both directions), and special configuration and programming are included.

Hence, when using closed-loop testing techniques, the results obtained include every aspect of the loop, and if the tool is powerful enough, even nonlinearities will be included

**FIG. 12.29**

Example of a PRBS-like test generated by having the operator move the SP. These data contain many rapid small disturbances from multiple sources (flow-path switching) in a clean-in-place (CIP) circuit. (From Matrikon TaiJi-PID.)

in the model. Nonlinearities are usually amplitude dependent. In addition, the amplitude of SP changes should reflect the usual process-variable excursions from SP.

Data Suitable for Closed-Loop Testing What is needed to have sufficient data to model a process? The first simple answer is: movements in the CO which are sufficient to cause process-variable movements which are larger than the noise in the system.

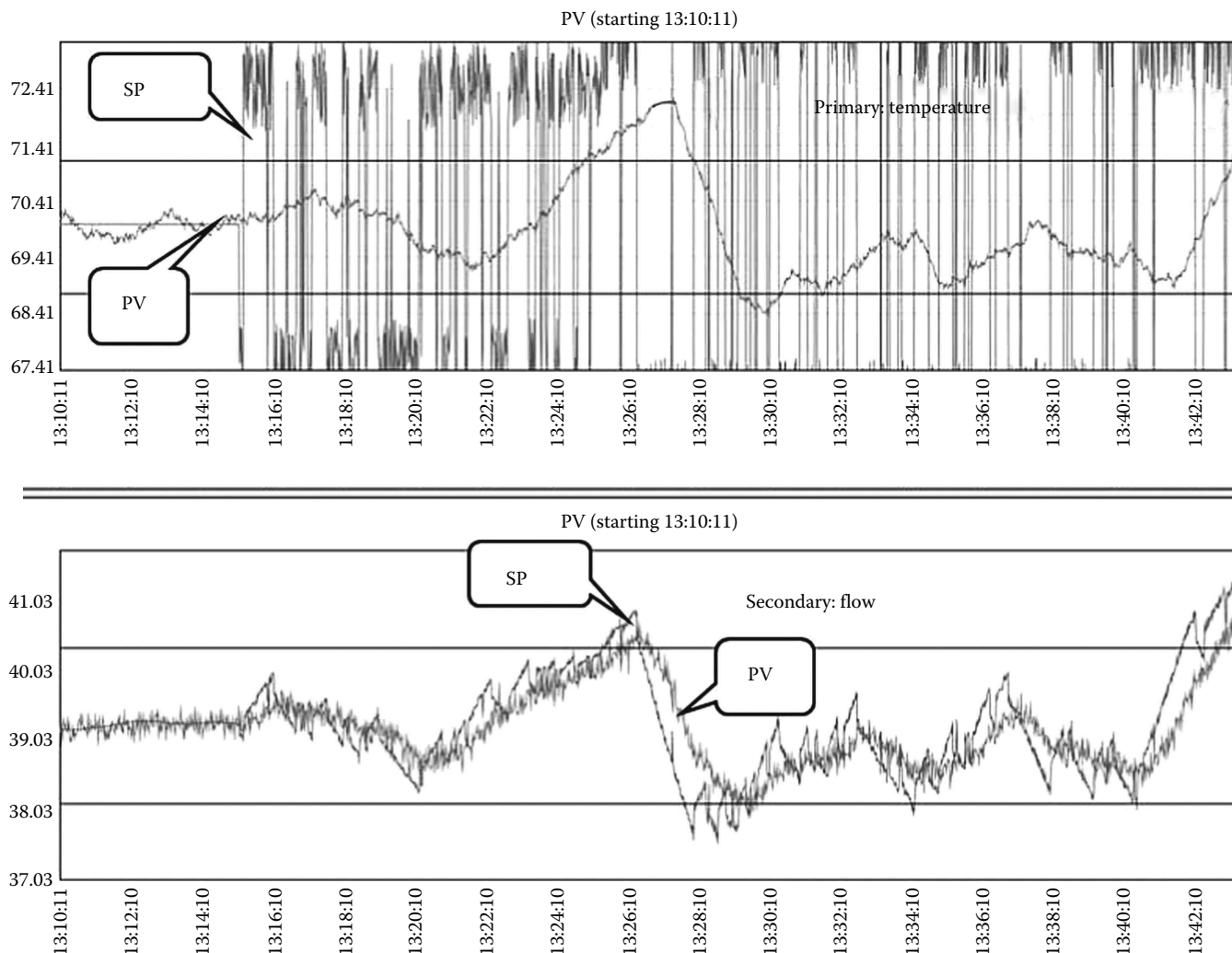
Benefits of Closed-Loop Tuning There are two key benefits to closed-loop PID testing. First, the PV can be kept close to normal operating conditions, resulting in less disruption to the process. Second, because the loop is in automatic mode, there is no need for dedicated personnel to spend time watching the loop closely, as would be the case in manual. In manual mode, a step test could result in a PV moving too far from its desired value. A disturbance could have the same effect when the PID is in open-loop mode. In automatic, a safe target can be set, and the loop has a good chance to recover from a disturbance.

If software is used that can generate a test signal for the SP movement, even less effort is required by engineering and operating personnel. For the software tests mentioned above, such as PRBS-like and GBN-type tests, the amplitude, width,

and duration of the SP changes can be safely specified to remain within desirable operating conditions. If limits are reached, tests can be stopped automatically, or SP changes can be forced to remain within limits. If operators move the SP during a test, the GBN will continue to be added to the new value. Multiple loops can be tested and tuned at the same time. It becomes feasible to tune all the loops in a plant.

When Closed-Loop Tuning May Fail When using closed-loop tuning methods, several conditions can lead to an unsuccessful test, for example, the CO hitting its minimum or maximum, a PV moving too far away from SP, disturbances occurring during a test, or special conditions occurring (pump stops, change in flow paths, etc.). Good tools may be able to handle these cases and exceptions. If the loop is in manual and has never been placed in auto or is unstable in auto because of current tuning or other problems, or if the PV is moving because of a disturbance in spite of a steady SP, then closed-loop tuning cannot be used. In the last case, even open-loop methods would not work.

When using closed-loop testing, the amplitude of changes used should represent situations similar to usual disturbances and changes. Indeed, if nonlinearities are present, tuning and modeling tools should handle these nonlinearities. Valve problems such as hysteresis and stiction, for example, will increase

**FIG. 12.30**

Example of an automatically generated closed-loop GBN test with a high-frequency dither on a cascade system. The master temperature-loop variables (SP and PV) are shown in the top plot, and the secondary flow output is shown on the bottom. (From Matrikon TaiJi-PID.)

identified dead time and result in more sluggish tuning parameters. Nonlinearities such as nonlinear gain or varying dead time will modify the real model, and a good tool should suggest more sluggish tuning to guarantee stability.

This example will be modeled using software, and an error bound will be calculated to include nonlinearities.

Tools Available for Closed-Loop Tuning Despite the sales “hype” for tools that “automatically” tune loops, the truth is that these tools often do not work as advertised. Some skills are still required to understand when these tools may or may not work; identification of these skills is one element of this paper. Next-generation tools will handle interacting loops all at once and model the interaction between these loops.

When analyzing a control loop, it is good practice to check whether the model varies with amplitude, direction, range (e.g., low flow and high flow), and process conditions. To detect nonlinearities in these relationships, tests must include various values of amplitude, direction, and range.

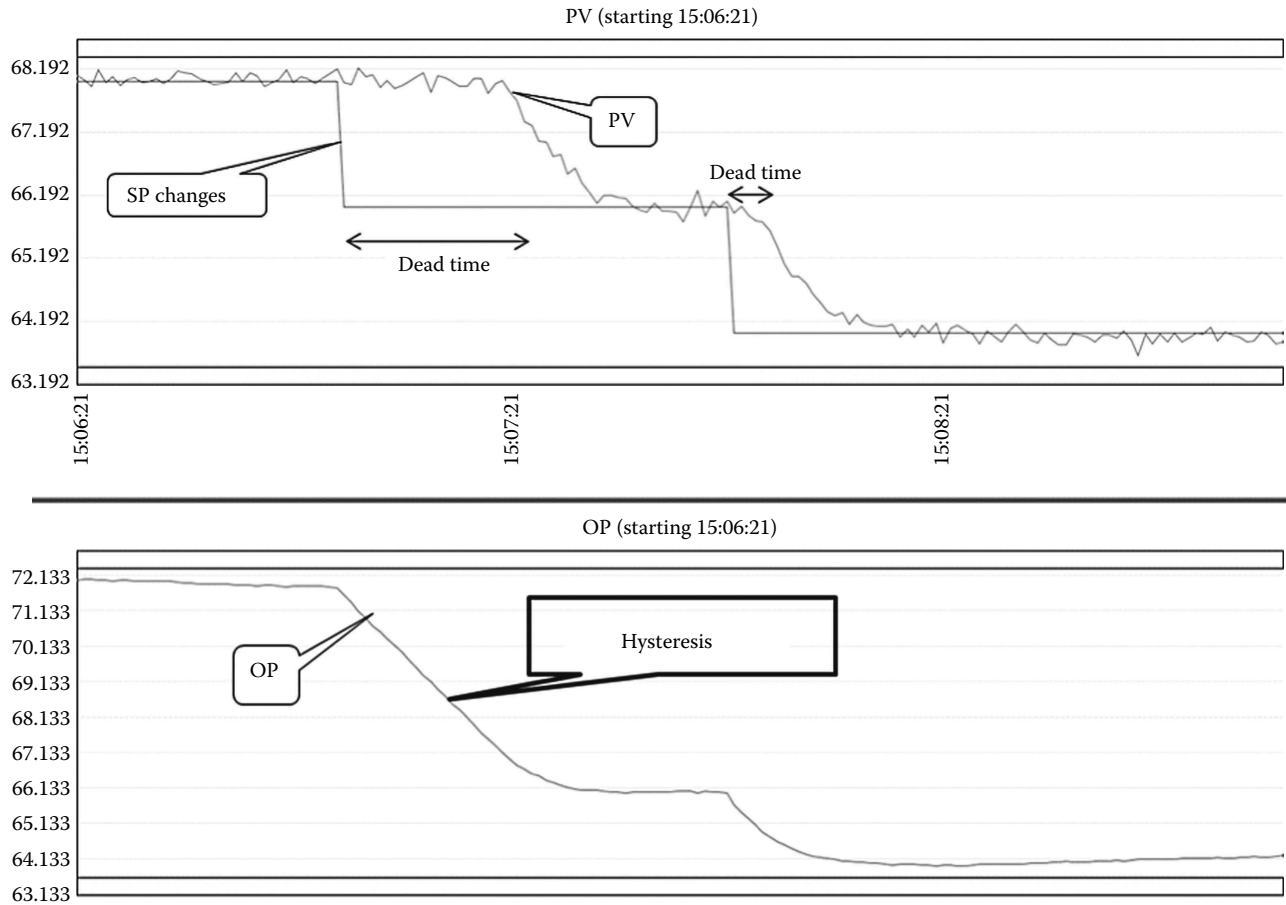
Most tools will then ask the user to analyze each test separately. After finding the worst case, the user will use this worst-case scenario to optimize the process and tune the loop.

With new tools, all tests are analyzed using a single stream of data, and the model is identified including an error bound to take nonlinearities into account.

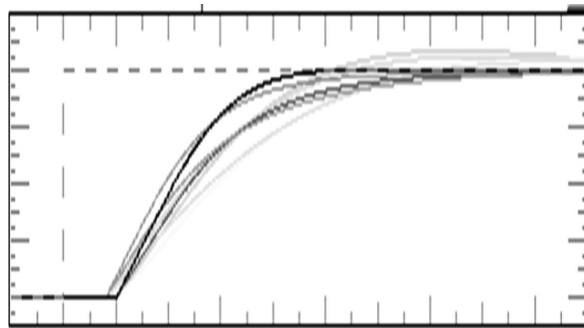
Handling Nonlinearities with Modern Tools If the model of a control loop varies, which model should be used for tuning? Nonlinearities and model changes are taken into account in new modern tools. Figure 12.32 illustrates different responses for the same control loop.

The worst case is (dotted line):

- Maximum dead time
- Maximum speed in closed-loop mode
- Maximum overshoot (process gain)

**FIG. 12.31**

Two SP changes for a flow loop. Note that the process dead time appears to be larger for the first SP change than for the second. This is due to hysteresis in the valve. (From Matrikon TaiJi-PID.)

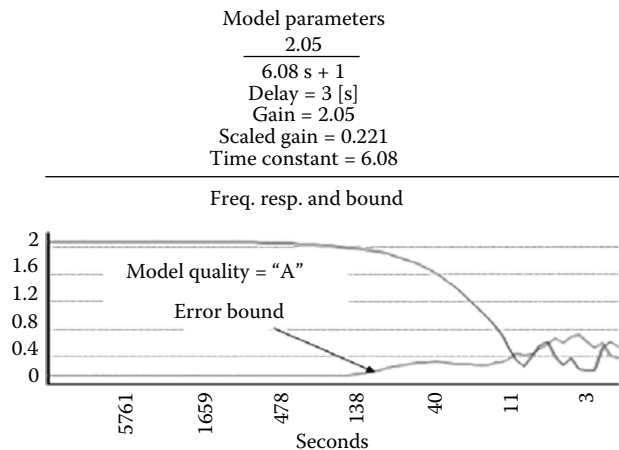
**FIG. 12.32**

Different responses for the same SP change. (From Matrikon TaiJi-PID.)

The worst-case model would be

- Maximum dead time
- Minimum time constant
- Maximum process gain

The software finds a model and determines uncertainty (nonlinearities and model variations). This uncertainty or error bound is displayed on a Bode plot (Figure 12.33) and

**FIG. 12.33**

Model derived and error estimated at all frequencies. (From Matrikon TaiJi-PID.)

represents for each frequency potential error of real process from the model identified.

When calculating tuning parameters, a robustness plot is displayed; this robustness plot illustrates how robust the tuning parameters are. Robustness is defined as tolerance to

instability. It depends on the tuning parameters, the model, and the error bound (uncertainties).

The distance between the graph and the horizontal line is the margin of instability at each frequency. For example, in Figure 12.34, the error bound is small at low frequencies (long period), but then increases at approximately 138 s. Finally, at high frequencies (short period), it is higher. When calculating tuning parameters, if an error bound is present, then tuning parameters have to be less aggressive to guarantee stability.

In the past, testing for nonlinearities (stiction, backlash, nonlinear gain, varying dead time, etc.) took much time and required a series of bump tests before tuning the loop.

The robustness plot is based on the model, the tuning parameters, and errors in the model (error bounds). Hence, the robustness parameter will include all defects. If the error bound is high, then the user can analyze the system in further detail only if the stability margin can be improved. Even in the presence of stiction, for example, the model will be correctly identified.

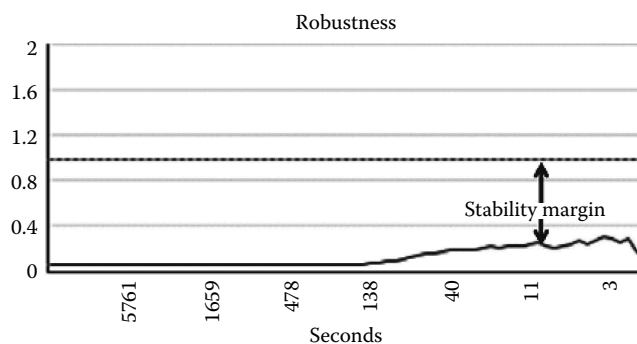


FIG. 12.34

Robustness calculated using tuning parameters, a model, and uncertainties. (From Matrikon TaiJi-PID.)

An Example A flow loop in which SP changes occur is used to demonstrate how to tune this loop without extra tests. This tuning is exemplified in Figures 12.35 through 12.37.

Very aggressive tuning on this loop can be observed. Most software will disregard these data and will be unable

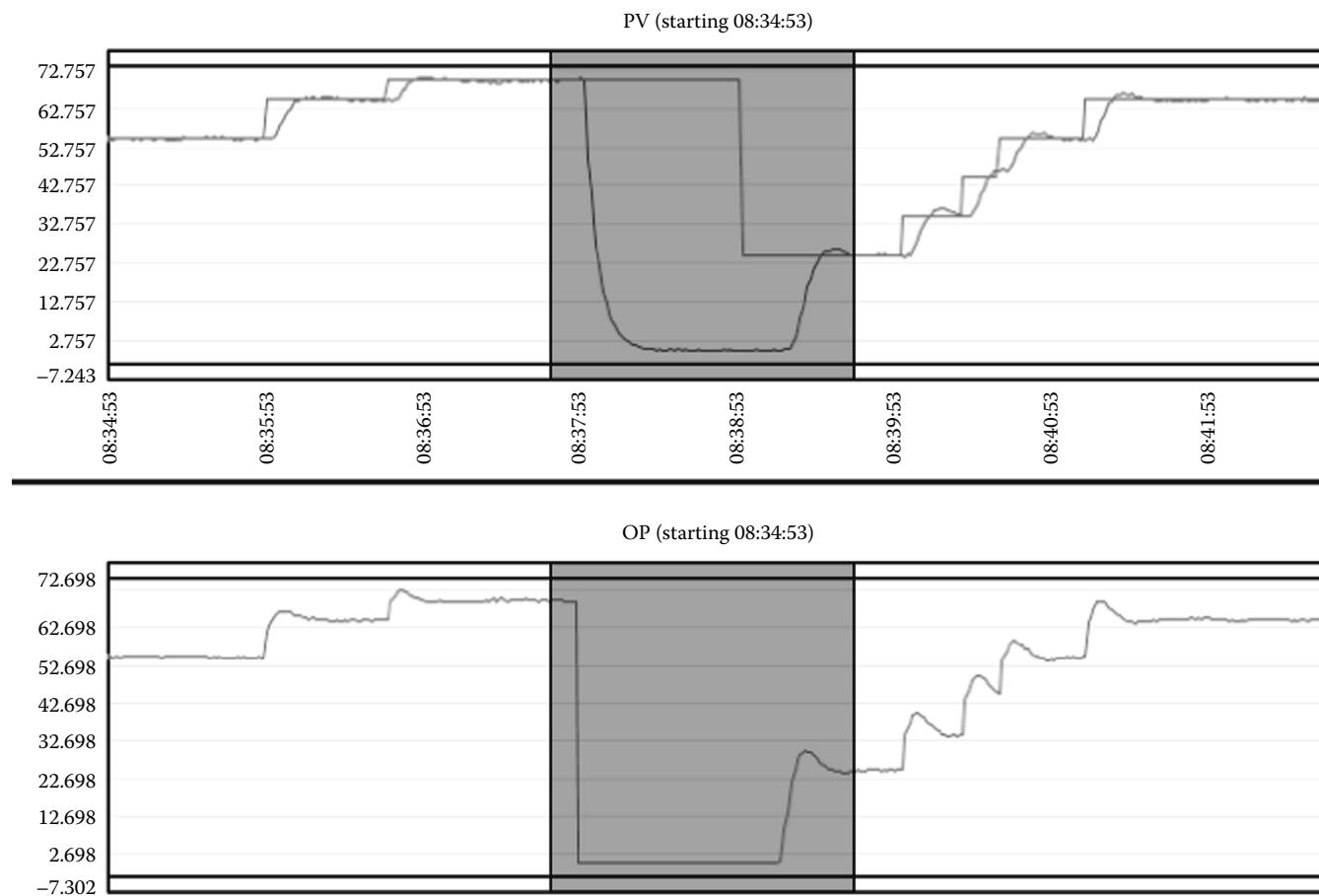


FIG. 12.35

Data used for analyses. (The gray area has been sliced out because the pump was stopped.) (From Matrikon TaiJi-PID.)

to model the process because oscillations and nonlinearities are present.

Even with nonlinearities, those tuning parameters will maintain this loop stable in all situations.

Summary of Closed-Loop Tuning Closed-loop tuning has gained popularity since new tools became available. With skills and experience, it is possible to use these tools to tune loops and optimize processes without interfering with production. The process control engineer can set up and configure tests in advance and let the program generate GBN with appropriate amplitude and frequency. Many loops can be analyzed simultaneously.

These tests can be done during normal production or during night shifts without process control personnel or intervention by operations. Operations can stop the tests at any time if they are uncomfortable with having the program manipulate the SP slightly around its value. It becomes realistic to tune all the loops in a plant because a campaign can be launched at any time to analyze a group of loops. Cascade loops, ratio control loops, and supervisory control loops can also be tuned while in closed-loop mode.

If the plant uses control performance monitoring tools, this software can identify which loops would benefit from

retuning and whether enough SP changes are already present to model and retune the loop using historical data.

All graphics have been generated using the Matrikon TaiJi-PID tuning tool.

Auto-Tune, Pre-Tune, Self-Tune, etc.

Manufacturers of industrial controllers (stand-alone, DCS, etc.) claim that their controllers are intelligent and self-tuning and are able to adapt themselves to the process.

The reality is often quite different. There are two different approaches to automatic tuning of controllers: Pre-tuning and Adaptive tuning.

To perform pre-tuning, a program tests the process and computes parameter tunings by analyzing the process response. The most common tests that these programs perform are either a bump test in open loop or an ultimate cycling test in closed loop. The latter test is called the relay method. A human being can perform the same tests and analysis in less time, and the results will be better. Moreover, pre-tuning programs frequently need user intervention and generate error messages.

To perform adaptive tuning, a program analyses the behavior of the process in automatic mode and computes

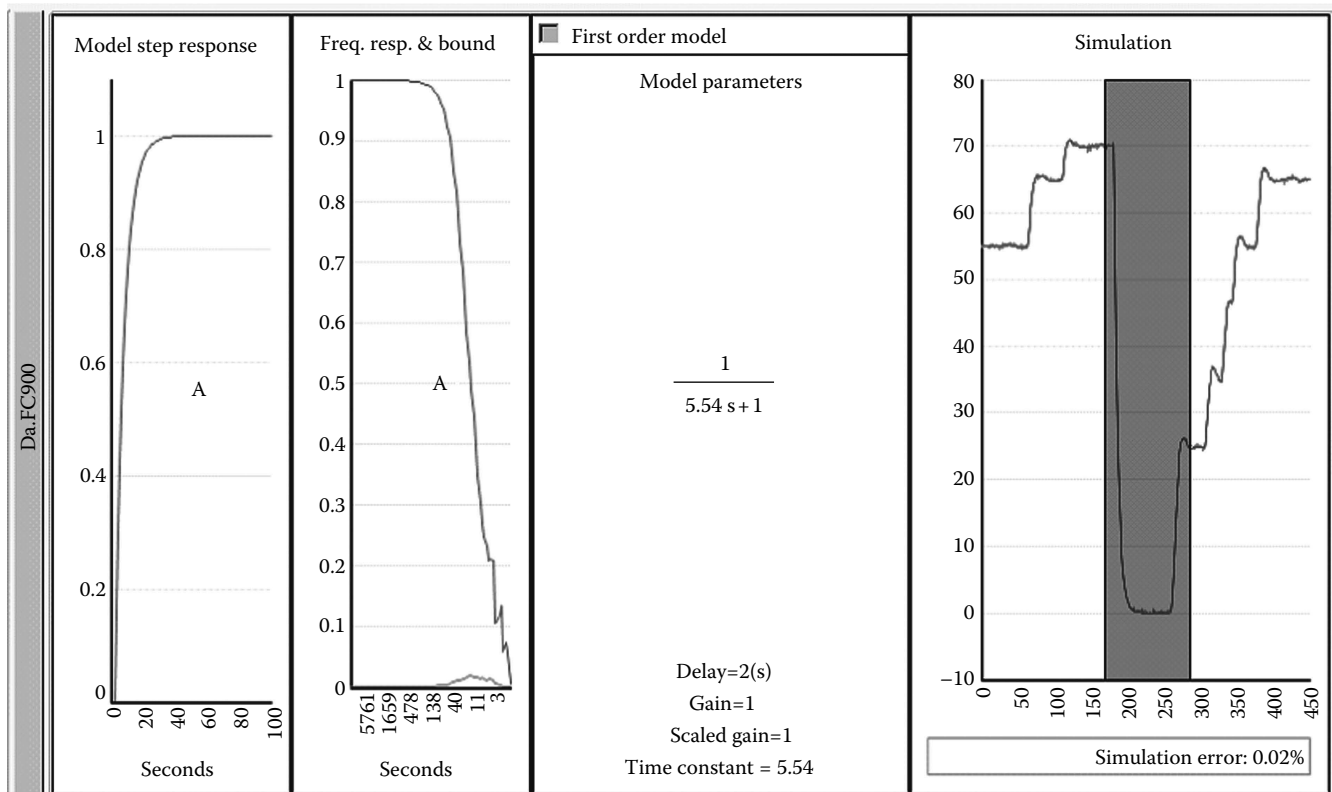


FIG. 12.36

Model as determined and uncertainty. Strong nonlinearities are present, but the software identifies a grade A model (high quality) with a small error. (From Matrikon TaiJi-PID.)

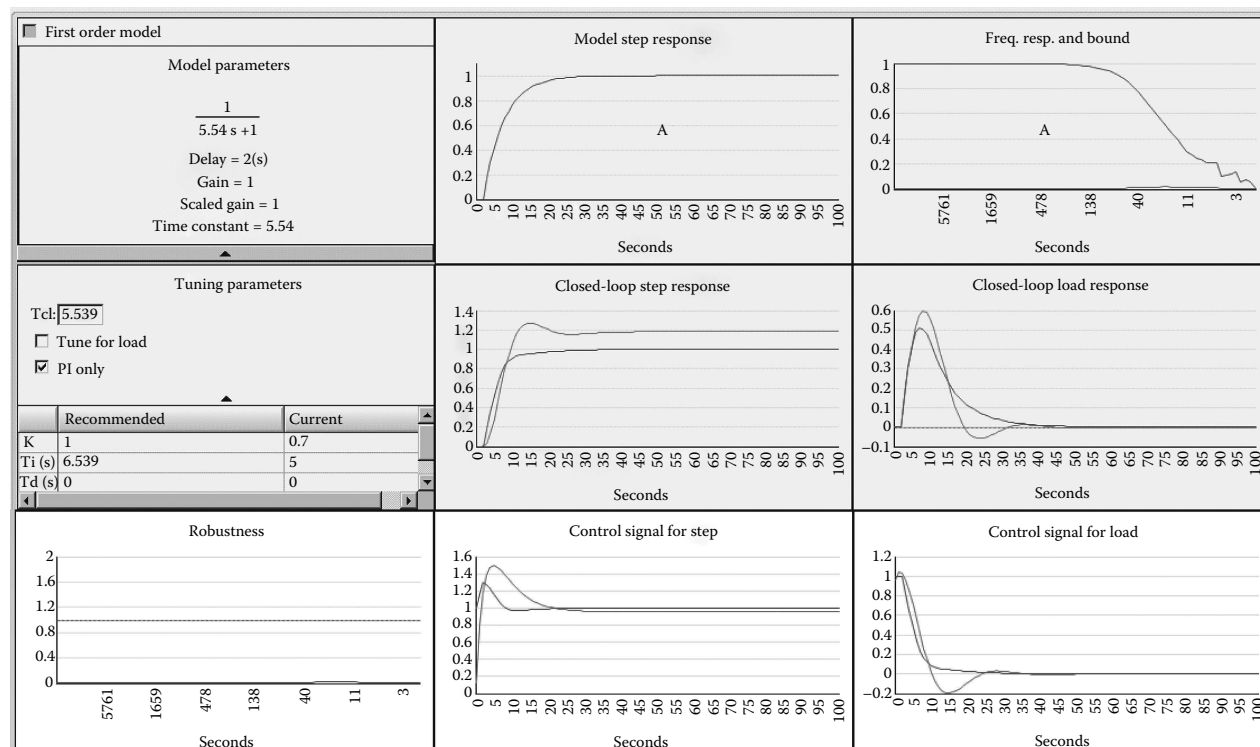


FIG. 12.37
Tuning pane and simulation. (From Matrikon TaiJi-PID.)

parameter tuning values. The program is based on heuristic rules or on complex mathematical models and formulas.

Programs that use heuristic rules are sometimes able to adapt correctly to the process, but they need supervision by experienced and skilled people. Frequently, these programs compute bad tunings, but if properly supervised (limits, pre-tuning, etc.), these bad tunings will not differ too much from initial tunings. The second method is based on complex formulas and elaborate mathematical models. These programs are rarely able to adapt correctly to the process, and they need to be supervised by experienced and skilled people.

Practical Considerations Both pre-tuning and adaptive tuning work well when the process model is constant, as in mechanical systems. If the process contains hysteresis, backlash, or stiction or is nonlinear, the algorithm behaves strangely.

Performance Monitoring A performance monitoring program identifies the areas in your plant where the greatest economic gains are possible it is designed to help you make the most impact on your plant.

The software automatically surveys any (or all) controllers in the plant and highlights candidates for performance improvement. Simple performance indices are used

to compare the performance of each controller. Additional information can be found in Chapter 11.

Bibliography

- Astrom, K.J., *PID Controllers: Theory, Design, and Tuning*, 2nd edn. Instrument Society of America, Research Triangle Park, NC, 1995.
- Brisk, M.L., Process control: Potential benefits and wasted opportunities, *Proceedings of the 5th Asian Control Conference*, Melbourne, Australia, Vol. 1, pp. 20–23, July 2004.
- Brittain, H., Dewey, D., and Ruel, M., Closed-loop tuning: Methodology, tools, benefits, and common mistakes, *Proceedings of the ISA 2009 Conference*, Houston, TX, October 2009.
- Brittain, H. and Ruel, M., Optimise your process using normal operation data, *Proceedings of the NPRA Conference*, Houston, TX, October 2008.
- Gerry, J.P., Tuning process controllers starts in manual, *InTech*, pp. 53–55, May 1999.
- Luyben, W.L., Luyben, M.L., and Bjorn, T.D., *Plantwide Process Control*. McGraw-Hill, New York, 1998.
- Patwardhan, R.S. and Ruel, M., Best practices for monitoring your PID loops—The key to optimising control assets, *Proceedings of the ISA EXPO 2008*, Houston, TX, November 2008.
- Ruel, M., How valve performance affects the control loop, *Chemical Engineering*, 107(10), C13–C18, 2000.
- Ruel, M., Loop optimisation: Before you tune, *Control*, 12(3), 63–67, 1999.

- Ruel, M., Loop optimisation: How to tune a loop, *Control*, 12(5), 83–86, 1999.
- Ruel, M., Loop optimisation: Troubleshooting, *Control*, 12(4), 64–69, 1999.
- Ruel, M., PID tuning and process optimisation increased performance and efficiency of a paper machine, *Proceedings of the 87th Annual Meeting*, PAPTAC, Book C, Montreal, Canada, pp. C63–C66, February 2001.
- Ruel, M., Stiction: The hidden menace, *Control*, 13(11), 69–76, 2001.
- Ruel, M. and Gerry, J.P., Quebec quandary solved by Fourier Transform, *InTech*, August 1998.
- Shinskey, F.G., *Process Control Systems*, 4th edn., McGraw-Hill, New York, 1996.
- Zhu, Y.C., Multivariable process identification for MPC: The asymptotic method and its applications, *Journal of Process Control*, 8(2), 101–115, 1998.